Bachelor Thesis

Title: "Modern regression methods with applications"

University of Macedonia
Department of Economics

Author: Charilaos Petrakogiannis,
eco17193

Supervisors: Ioannides Dimitrios,
Panagiotidis Theodoros

Contents

Acknowledgments

In the following section I would like to express my deep appreciation to Professor D.Ioannidis and Professor T.Panagiotidis, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to thank Laboratory Assistant I.Athanasiadis, for his precious advice and assistance in keeping my progress on schedule and helping me overcome any difficulties that had to do with the coding part. Their help was significant for this research and I feel blessed for having them by my side.

Also, I wish to acknowledge the help provided by prof. Pavlidis Georgios (University of Patras) for giving important references about the topics and last but not least, I would like to extend my thanks to my friends and family for their constant support throughout my academic life. I owe them a lot, because I wouldn't be the person I am today without them in my life.

"There are going to be times when you learn more about the world you're entering and feel defeated when you see the gap between the ideal and the reality... But that's something we'll all face. The people that face those obstacles and overcome them are people whose dreams come true."

Tsugumi Ohba

Introduction

Nowadays more and more scientists, corporations, employees and academic students around the world use statistical analysis and various regression models in order to analyze different datasets and come to conclusions that will help them advance in their jobs or even in their lives.

Just like all these people, in this specific thesis report, we are also going to use statistical tools like R programming language in order examine eight different regression methods (multiple linear, ridge, lasso, polynomial, logistic, poisson, decision tree and random forest) over a dataset that contains information about a variety of Portuguese red wines. Through this pioneer project we have faith that we'll find out the most suitable regression type/s for a wine quality analysis, but also for a discrete dataset analysis generally.

In the following sections we will present and analyze the theoretical background for each one of the eight different prediction methods, a significant amount of literature review that exists at the moment and which is dedicated to the practical application of these models over a variety of real-life sectors and of course we are going to compare the statistical results that we are about to receive from the usage of Rstudio and that will help us succeed in our goal, shedding light on which is the most appropriate regression type for this specific task.

Literature review by examples

In this section we are going to present and discuss the progress that has been made in every regression type (from the ones that we study) through the ages in order to record and categorize their characteristics and field of work. For this reason we will examine a variety of scientific papers from all over the world which are considered crucial in data science.

Multiple linear regression

Considering MLR there are many research results and studies that have examined its effect on a variety of datasets. This is something logical if we take into consideration that linear regression is the most common and applied regression type in the world.

In 2007, City University of Hong Kong published a research (Tso and Yau, 2007) in which three different regression types (MLR, neural network and decision tree) were compared in order see which one was the most appropriate and accurate to predict the future electricity energy demand in Hong Kong. These three different models were applied on a dataset that was made back in 2000 and included the energy consumption of the domestic households according to their housing type, characteristics and appliance ownership in two different seasons (summer and winter).

In order to compare these three models, the scientists set RASE as the major criterion for this job. After having conducted the befitting applications for each regression type, the results showed that MLR wasn't the most suitable for this energy consumption prediction, because both neural network and decision tree showed better RASE values than the linear model (neural network model had the best performance in the winter phase and decision tree model had the best performance in the summer phase).

Critique: Although MLR wasn't the best model in order to predict the energy consumption it wasn't bad at all and its RASE values were very close to the remaining two regression models showing us that in general was able carry through with this prediction task, but the selection of the right prediction metrics in addition with the complex form of the dataset, resulted in these specific outcome that was in favor of neural network and decision tree regressions.

Later in 2015, Gazi University of Turkey having noticed that there were a boom of new regression methods over the last decades, tried to predict the electricity consumption in the long term (Kaytez et al., 2015) adding the LS-SVM method in a prediction package that already included MLR and neural network. These three regression methods were applied on a dataset which included data from 1970 to 2009 and was collected from several national organizations (TEIAS, TIE and TEDAS) and private businesses that were related to the energy sector. Installed capacity, gross electricity generation, population and total subscribership were part of the independent variables for every single model and the major criteria that were setted in order to measure

their performance were six (MaxError, MAPE, MSE, RMSE, SSE and R-squared).

All these previous criteria ended up being insufficient to measure the validity of the three models and for this reason the scientist conducted a ROC analysis to the net electricity consumption of Turkey in order to count the sensitivity and specificity values (of the three regression models). This ROC analysis concluded that the LS-SVM model was more sensitive than the others and considering specificity the LS-SVM and neural network methods had better results than the linear regression.

In conclusion, the LS-SVM model seemed to be the most appropriate for an electricity consumption prediction, surpassing the remaining two methods (MLR and neural network) by 1.70% and 0.88% respectively.

Critique: This research was able to show that even if the linear regression is the most common and widely known prediction method around the world, it has become one of the least reliable and accurate ways for long-term energy consumption forecasting. Furthermore, the traditional ways of evaluation, like the ones above, tend to be less useful in pioneer analyzations in sectors like this (electricity) and tools like ROC curve or RASE, in the previous example, provide better results with more complete information when the dataset comes with a variety of advanced information.

Moving on 2019, IEEE (Institute of Electrical and Electronics Engineers) presented a research based on the prediction of graduate admissions (Acharya et al., 2019**).** In this scientific analysis four different types of regressions (MLR, support vector, decision tree, and random forest) were applied on a dataset based on various student profiles (marks, tests, recommendations and personal choices) in order to find out the best university choice for each postgraduate student and compare these results with the primary choices of each student. Through this model analysis, the four regression forms and their statistical errors were compared to each other in order to end up with the most successful, considering prediction performance, regression type that would be able to achieve the purpose above in the most accurate and reliable way. For this reason the analysts set MSE and R-squared as the two major criteria in order to evaluate the performance of each regression method.

After the appropriate preparatory work, the results showed that MLR, support vector, decision tree and random forest regressions had all low error values (MSE). Although, multiple linear regression was able to achieve both good data concentration (R-squared) around the fitted regression line and low error values, surpassing the values of the remaining three methods.

Critique: These results made clear that MLR is a very reliable form of regression for predictions in quantitative datasets, especially when these are described by linear relationship of features.

MLR Critique: To sum up, the previous three scientific researches made quite clear that there are some sectors in which the linear regression model is the most reliable form of forecasting (situations in which the datasets are described by linear relationship between its components) and some others in which there are different and more accurate ways that we should adopt in order to predict future values (for example the energy consumption sector). Nevertheless, it has to be said that MLR after all is a regression method that's

a benchmark in many scientific researches and will always be a yardstick of prediction performance comparison for different regression models.

Ridge and lasso regressions

As we already know, over the last years many new regression types have been developed and evaluated in order to become a useful tool for data science. Two of these new methods of prediction were ridge and lasso regressions which are part of the shrinkage methods (a regression method that we will discuss in the following chapters). This need for new prediction models made many analysts to study and research these particular types of regressions meticulously in the last decades, providing us with a variety of scientific papers that have to do with the classic, but also with their evolved forms.

Back in 2000 professor Edward C.Malthouse from Northwestern University's Journalism School compared ridge regression method with variable subset selection (VSS) in order to see if there would be an improvement in the direct marketing scoring models and their performance (Malthouse, 2000) These models were crucial for building a trustworthy and a profitable relationship between the company and the customer, so in order to succeed his goal he applied these two prediction methods on a dataset made of information that was collected from the Direct Marketing Educational Foundation (DMEF) and contained data from a multidivisional business.

The analysis of these two methods showed that ridge regression had after all a better prediction performance compared to VSS. Ridge regression performed better than subset selection in many different variable sets and especially when the model had a big number of explanatory variables. Furthermore, RR seemed to be more stable than VSS in its prediction estimations.

This research made clear that ridge regression can be a very useful tool in predicting the customers' behavior regarding their spending frequency inside a corporation, showing us that this form of shrinkage method is more preferable in the marketing sector than the traditional OLS methods.

Critique: It seems that ridge regression is more useful and provides better and more stable estimations in situations where the dataset contains a big amount of explanatory variables in comparison with different prediction methods that apply subset selection in order to make predictions.

Years later and specifically in 2016, a team of scientists from many universities around the world tried to study and evaluate the prediction performance of a specific ridge regression type, the Kernel ridge regression (Exterkate et al., 2016). The uncertainty that already existed in many complex finance forecasting models, due to the variety of their variables, made them think that it was time to end up with a model that could cope with high-dimensionality and nonlinearity at the same time. For this reason they compared KRR and several principal components models that were applied on a data-rich environment (of macroeconomic and financial parameters like industrial production, personal income, manufacturing & trade sales and employment) that included information from 1970 to 2009.

After having conducted the basic regression steps and comparing a variety of prediction metrics (mean, PC, Gauss, Poly etc.) the results of the research showed that a transformation of the original KRR, in order to be able to

predict the variables well, even with time series, were very successful. KKR was in many ways, a better forecasting method than every other non-linear regression, benchmark or Principal-Components (PC) method.

Critique: KKR forecasts showed more information than other non-linear forecasting methods (including threshold autoregressions, polynomial sieves, nonparametric regressions, and non-linear principal component regressions) making crystal clear that this kind of ridge regression is a useful tool for any macroeconomic or financial forecasts where the information is very complex and is not easy to be handle from many tradition linear or non-linear prediction methods.

Considering lasso, the start was made in 1996 when Robert Tibshirani from University of Toronto conducted a research on this specific prediction method (Tibshirani, 1996). First of all, he tried to present this new prediction form by comparing it to ridge and subset selection methods making clear that this new model is capable of minimizing its coefficients until they are exactly zero. Lasso regression was able to sustain the positive characteristics of ridge and subset selection models and at the same time it could wipe out the drawbacks of the previous methods (these drawbacks had to do with the incapability of having coefficients that were exactly zero and the significant variability in the prediction results of subset selection, due to its non-continuous form of process). In order to analyze in a better way this new model, he provided its theoretical background and he applied it on a prostate cancer dataset that was obtained seven years ago from another study along with other two prediction methods (OLS and subset selection).

The results confirmed the theory and lasso, indeed, had driven some of the model's coefficients to zero. Nevertheless, Tibshirani hadn't examined lasso's prediction accuracy yet. For this reason he conducted a series of simulations on fifty different datasets that were applied on five dissimilar prediction methods (OLS, lasso, ridge, subset selection and non-negative garotte). The conclusion was interesting, because lasso regression showed to be the most suitable prediction method (according to MSE) in cases where the average-sized effects were ranging from small to medium numbers.

Critique: This pioneer study had done something remarkable. For the first time in history, Robert Tibshirani shed light on this new statistical chapter, making clear that lasso regression was a very useful prediction tool, especially when some of the dataset's variable were totally insignificant for the prediction procedure and their inclusion in the final model could lead to wrong conclusions.

Sixteen years later, Research Institute of Economic Statistics and Quantitative Economics of Zhejiang Gongshang University used the lasso and the support vector regressions on a dataset that was taken from China's Undergraduate Mathematical Contest in modeling (2012), in order to evaluate the quality of some Chinese red and white wines and to compare the performances of these two prediction models over this specific task (Yao et al., 2017). For this reason the researchers set MAE and MRE as the two major criteria that would be considered as the main comparison metrics between the two models.

After the completion of the appropriate code functions, the two models resulted in eight physicochemical indicators that are responsible for the wine quality. Considering the prediction performance of the above mentioned methods, lasso regression and SVR was both a good prediction selection because the first one was used in order to check the linear relationship between these eight explanatory variables and the second one helped in order to study the nonlinear effect of these eight effective components. The results of this study showed that if the training samples are less than ten, then according to MRE the SVR method seems to be a better decision than lasso, due to the fact that when there is a small training sample the nonlinear effect of the indicators affects more the prediction estimations than the linear one.

Critique: In conclusion this research was able to examine and evaluate, in a successful way, lasso and support vector regressions and concluded that both models are able to make good and accurate predictions depending on the sample size of the training set which is the main factor that determines which one is the most suitable prediction model for wine quality evaluations.

Moving on last year (2019), three university professors banded together in order to work over a project that was similar in many ways with the one above. In this last scientific research, three types of regression (OLS, ridge and lasso) were compared in order to find out which one is the best prediction tool for a wine quality classification (Mayooran Thevaraja et al, 2019). These three regression forms were applied on a dataset that included all the eleven components that can be found in white wines in north Portugal and classified the wine categories from zero to ten depending their quality. The major criterion that was chosen for the purpose of prediction performance comparison between these three models was RSS.

Considering ridge and lasso, both of them shrank their λ parameter close to zero, but only lasso were able to reach it making clear that depending the λ value the model would include ten or six independent variables and when it comes in the linear regression only three variables were statistically insignificant. After having conducted their predictions in training and testing sets, they concluded that the RSS value in training test were better in the linear regression due to the lack of coefficient penalties, but in testing set the ridge regression had the minimum RSS followed by the one of lasso regression.

Critique: In conclusion, this research managed to make crystal clear that regularized models like ridge and lasso are best fitting models in regression analysis when noises exist in the usual models, in comparison with OLS which seems to be problematic with its prediction estimations when there is a big data and when there is differentiation between the training and the testing set.

Ridge and Lasso Critique: All these different scientific analysis succeeded in shedding light on the prediction methods of ridge and lasso regressions. Each one of them was applied in various sectors of data science like marketing, wine quality and finance forecasting in order to produce prediction estimations. For that reason ridge's and lasso's prediction accuracy was compared to the one of other methods (PC, OLS, VSS and SVR) and turned out that these two models were suitable for predicting the future

values when the datasets had a big number of explanatory variables and noise existed, due to the fact that they could easily distinguish which components were crucial for the prediction excluding every other factor that could lead us away from the right conclusions and results.

Polynomial regression

Polynomial regression is our next prediction method. This nonlinear type of regression seems to be used by many scientists and analyst from all over the world in order to predict and compare research analysis. In this section we are going to discuss its significance in different scientific sectors, its benefits and drawbacks and we will evaluate its prediction performance accuracy through two important research papers.

In 2006, Mitsubishi Electric Research Labs compared and examined two different prediction models (locally weighted regression and polynomial regression) in order to diagnose any possible fault detection in their air-cooling systems ("A Comparison between Polynomial and Locally Weighted Regression for Fault Detection and Diagnosis of HVAC Equipment - IEEE Conference Publication."). Therefore, they applied these two regressions on a dataset that were collected from their HVAC systems that were operating under normal conditions and included sensor measurements of the external and the mixed air temperature from two different coordinates during two different chronical periods. For this model comparison, the analysts compared the accuracy of three classifiers that could tell apart the different charge states of HVAC (normally charged, overcharged and undercharged equipment) and were built using the residuals of the above mentioned prediction models.

After the accomplishment of the right functions, the results showed prediction accuracy differences between LWR and PR. To be more specific, the three classifiers were able to distinguish more easily if the air-cooling systems were operating under normal or faulty charge conditions when the residuals were coming from LWR. Polynomial regression were able to end up with an accuracy percentage of 78% while LWR had a percentage of 95%.

Critique: In conclusion, polynomial regression were inferior to locally weighted regression in this specific case of fault detection identification and this can be explained by the fact that LWR is a prediction method which is more flexible than the polynomial one. This characteristic of flexibility can be very crucial wherever there are several local dependencies inside a dataset and in this case helped them a lot to choose the most suitable prediction method between these two options. Nevertheless, unlike many other researches, this one used classifiers as the major comparison tool and didn't check for any other prediction metric that could have resulted in different conclusions for each one of these two models.

Eleven years later, in 2017 a group of engineers tried to compare the classic linear regression with the polynomial curve fitting model which is an extension of the original nonlinear method. They wanted to forecast the energy production until 2030 by applying these two prediction types on a dataset that was made of consumption and production information over electrical energy in Morocco that was collected during a period of sixteen years (2000-2015) (Kafazi et al., 2017). For that reason they set R-squared and adjusted R-squared as the two major metrics that would be responsible for this model comparison.

After having conducted the right functions on this specific Moroccan dataset (this selection was made due to the fact that Morocco has replaced a big percentage of its fossil fuel based electrical production model with renewable energy systems) the results were able to show that the polynomial of the 2nd order, which was the best choice among other options of polynomial degrees, showed high R-squared values making clear that is a good tool for demand-production fluctuation predictions in the energy sector. On the other hand the linear model, that's been a milestone in energy forecasting for many years, had lower R-squared marks than the previous method.

Critique: In a nutshell, polynomial regression seems to be a significant forecasting instrument for the energy production-consumption sector and it is possible that in time it could be able to displace classic prediction models like OLS. Although, it has to be said that the lack of variety in the prediction metrics has showed in many cases that the prediction accuracy results can differ and in this study only two of them have been adopted.

Polynomial Critique: To sum up, through this two-paper analysis we were able to tell that polynomial regression and its extensions seems to be widely popular, especially into the engineering sector. This model has been a good choice wherever there is a nonlinear relationship between the regression components (variables), but it is not quite clear yet in which cases it should be considered as the most suitable option without comparing it first with other and more widespread methods.

Logistic and poisson regression

Logistic and poisson regressions are members of the generalized non-linear regression models family. Each one of them has been tested and applied in numerous scientific sectors proving that both of them can be very helpful in different situations, especially when the task has to do with the prediction of probabilities.

To be more specific, in 2002 the Upper Austria University of Applied Sciences tried to analyze and compare logistic regression and artificial neural networks using a sample of 72 different scientific papers (Dreiseitl and Ohno-Machado, 2002).

Through this analysis they concluded that both models are forming their parameters using maximum likelihood estimation. The logistic model tends to be steadier than ANN which is more flexible and for this reason it is more possible to face overfitting situations in their parameter prediction, but in both cases this can be avoided by variable selection (forward, backward and stepwise method for LR & automatic relevance determination or sensitivity analysis for ANN).

In order to evaluate the prediction accuracy of these models, scientists use a variety of methods, but in this research the two main criteria are discrimination and calibration (the first one measures the successfulness of data separation into two classes and the second one measures the accuracy of probability estimation). After having conducted a variety of different methods that aimed to produce the best results for these two criteria, they concluded that both models perform more or less the same, but ANN has better results on the discriminatory power.

Critique: To sum up, both logistic and artificial neural networks have numerous similarities and differences but in general they tend to produce reliable and accurate results for many scientific sectors including the healthcare sector which is the most common performance area for LR and ANN. Last but not least, this academic analysis showed the majority of scientific researches uses more often the logistic model rather than the ANN, due to the fact that the first one is easier to build and provides simpler information that can be understood in a better way than the one of ANN.

Seventeen years later, in 2019, a group of scientific institutes from China tried to move on even further and compare logistic regression with twenty-two machine learning models in order to evaluate their performances on predicting the possibilities of survival after a serious brain injury (Feng et al, 2019).

For them to accomplish this evaluation, they applied both logistic and machine learning models on a medical dataset that was collected from Sichuan Provincial, China during 2009-2011, which had information for 117 patients, and compared them using seven different metrics (Accuracy, F-test, ROC, AUC, precision, recall and decision curve analysis). Their results showed that only the F-test score (0.93) of the logistic regression was slightly better than the one of the machine learning models making clear that the second ones were more capable of predicting the patients' survival possibilities.

Critique: This tremendous discovery made them notice that even if the logistic prediction model has proven itself very useful and helpful in scientific sectors like the medical sector, the machine learning algorithms can overcome its performance in some cases like this one, due to the fact that are able to handle more easily high-dimensional factors that can be found in this kind of datasets, making this prediction model an appropriate way to embody more risk factors for prediction.

A year later University of Naples, Italy tried to accomplish one of their researches using a poisson regression model instead of a logistic or even a multiple linear one. In fact the university's engineering sector applied a geographically weighted poisson regression model over a dataset that was made of information that had to do with 99 Italian provinces and their tourist arrival rates, but also their transport models during the years 2006-2016 in order to see if Italy's High Speed Rails (HSR) had any impact on tourism levels (Pagliara and Mauriello, 2020).

After having conducted their regression analysis the results were remarkable. It seems that GWPR (Geographically Weighted Poisson Regression) was able to shed light on the dependence between tourism and HSR. To be more precise, the research showed that HSR were responsible for the increase of tourism in areas where the rail system was highly developed and the decrease of tourism where the opposite was happening, but also it showed that the biggest change came from the foreign side of tourism, due to the fact that foreigners were more familiar, than the Italians, with this kind of public transportation and so this system helped them to visit the Italian provinces that were connected with this. These results showed, that geographically weighted poisson regression was the most sufficient prediction model to conduct such an analysis.

Critique: The main reason is that datasets that contain geographical information during a continuous amount of years, tend to show signs of spatial autocorrelation and unobserved heterogeneity that prevent us from having reliable prediction results. For this reason, the fact that this analysis managed to overcome this specific problem, made crystal clear to everyone that this regression type is the most suitable for tourism planning and transport investments.

Logistic and Poisson Critique: It seems that both logistic and poisson regressions tend to have their ups and downs. Considering logistic regression, it is well-known that it is very easy to be build up and to provide its predictions very easily in comparison with other regression methods, but the fact that it cannot handle well high-dimensional information makes this method unreliable for complex sectors like the medical one. On the other hand, poisson regression seems to be the right tool for predictions on datasets that contain variables with autocorrelation and heterogeneity, but this kind of data isn't so usual, putting a lid on the variety of sectors that can be applied on.

Decision tree and random forest regression

Finally, in this paper we are going to discuss about a unique category of prediction methods. This category includes the last two regressions (that we will analyze in this thesis report) that are called decision tree and random forest and belong to the tree-based prediction models family. Both of them have been applied to a humongous variety of papers due to their "different" way of predicting possible results and to be more specific we are going to take a look at three scientific papers in which the scientists have shown their prediction performance over a variety of statistical problems.

Firstly, in 2011 a variety of American and Chinese universities and bank institutes cooperated in order to accomplish a common goal. They tried to find out if they could predict the amount of the bank customers that would become churners during the next few years, so the banks would be able to keep them active and achieve a better level of Customer Relationship Management (Nie et al., 2011).

For them to accomplish this, they used two different prediction methods, the logistic regression and the decision tree regression which applied over a dataset that was obtained from an anonymous Chinese bank that included information about 60 million bank customers and their credit cards (information of the card holder, basic information of the card, detailed transaction information, abnormal usage information of the card, etc.) over a timeline of 3 years (2005-2008). Furthermore, they adopted Percentage of correctly classified (PCC), receiver operating curve (ROC), top-decile lift and Gini coefficient as their four major criteria in order to evaluate the prediction accuracy of the models.

The results showed that these four criteria that were mentioned above weren't enough in order to end up with reliable results, due to the fact that they check only the accuracy of the model, without considering the economic cost of the prediction. For this reason the scientists added another criterion, the one called misclassification cost that was responsible for overcoming the previous problem. Finally, after evaluating 135 different variables, they kept 95 of them as statistical significant ones and the best model (which was a logistic one) had only 15 variables in each regression body. Last but not least, the comparison of logistic and decision tree regressions, have shown that the first one was a better choice for this specific task, because decision tree didn't check for multicollinearity which was a major criterion in ending up with statistical significant variables or in other words, decision tree regression wasn't able to end up with an amount of variables that were valuable and significant for their prediction model. However, decision tree regressions are able to provide useful information for bankers, because they have the ability to predict easily the future customer decision wherever the only information that they can obtain is about the customers' transactions and not their personal characteristics.

Critique: Just like many other scientific papers, the right selection of prediction metrics seems to be the major key in a regression comparison. This

time the extra addition of a metric that measures the economic cost and embodies it on the final prediction results, made clear that the accuracy isn't the only thing that matters in statistics. Furthermore, the decision tree's inability of measuring the multicollinearity among the dataset's variables seems to be a major problem in situations where this can cause a false variable separation in significant and insignificant ones.

However, six years before this aforementioned analysis, the department of marketing of Ghent University, tried to examine the importance of Customer Relationship Management, for a company, comparing the traditional set of linear and logistic regressions with random forest regression which belongs as we already know to the tree-based prediction models family. In order to complete this task, the department of marketing applied two different forms of this regression (binary classification and regression forest) over a dataset that was obtained from the data warehouse of a large European financial services company and contained 100.000 observations that were account for the same amount of firms' customers. They ran the same models with four different dependent variables (next buy, active partial-defection, profit drop and profit evolution) and three types of explanatory variables (past customer behavior, observed customer heterogeneity and variables related to intermediaries) and tested their prediction accuracy using the AUC and MAD criteria (Larivière and Van den Poel, 2005).

The empirical study showed that both random forest techniques ended up with better predictions compared to the ones that linear and logistic regressions brought up. To be more specific, random forest that had partial-defection in the position of the dependent variable had the best accuracy stats, but it is worth mentioning that the independent variables had a different impact in the four different types of the same model. The scientists ended up with the conclusion that past customer behavior is an explanatory variable that is more likely to result in a better repeat purchasing and favorable profitability than the other variables, while intermediary variables have an advanced impact on partial-defection.

Critique: This great analysis made clear that in cases where the dependent variable consists of a subgroup that contains more than just one variable, we'd better use random forest regression and not the traditional models of linear or logistic regressions, if we want to end up with reliable and accurate prediction results. In other words, wherever the statistical problem tends to need a simultaneously analysis of different dependent variables, random forest regression is the most sufficient type of prediction method for this job.

Last but not least, the final scientific paper is about an interesting statistical analysis that was conducted in 2014. That year two different schools from the University of Southampton (Electronics and Computer Science & Management School) cooperated in order to examine the effectiveness of random forest regression in building a well-functioning trading strategy by predicting the price return over seasonal events that have been documented in financial data for the last few decades. They used this tree-based prediction method because the traditional ones even though they are

very effective in many ways, they result in profit decline in many cases. For this reason they applied a random forest regression over a dataset that was collected from the DAX stock index and contained information about 30 different stocks during thirteen years (2000-2013) and compared this trading system with every other that was previously been in use, using a variety of prediction metrics (MD, MAPE, MSE) (Booth et al., 2014).

The analysis showed that random forest regression was the best one and outperformed every other traditional method that was tested (linear regression, decision tree, MLNN, SVR). To be more specific random forest managed to have the lowest scores in MD, MAPE and MSE tests in the testing dataset, resulting in better prediction and profitability accuracy.

Critique: This result made clear that random forest regression is a useful tool in situations where the dataset contains financial information and more specific, information that varies from season to season, overcoming that way other traditional methods that do not give the appropriate attention on this informational interchange.

Decision tree and Random forest Critique: This analysis in addition with the ones that were mentioned above, succeeded in giving us a crystal clear view of how decision tree and random forest regressions work. These two very important tree-based models are capable of ending up with very reliable and useful predictions wherever there is a problem that needs desperately a simultaneously analysis of various dependent variables that can't be done efficient with traditional methods and for this exact reason these types of methods have an enormous success in the banking and financial sector, where the need of classification analysis is more usual to be found.

Chapter 1

Dataset analysis

Theoretical background

In the following section we are going to present the analysis of our red wines dataset through a variety of histograms and boxplots that describe in a scientific way our dataset's characteristics. Also we are going to conduct a correlation analysis of the dataset's variables through an array of plots, including a PCA analysis and we will test the normality of our dependent variable with or without its transformation.

Considering the second part of our dataset's description, correlation is about the statistical relationship (positive or negative) between the variables of a model and can be found with several different ways and functions in R (plots, tables, pies etc.). Furthermore, correlation is connected a lot with the PCA analysis which is a statistical method that uses the basic components of a dataset (James et al.). Principal Components Analysis (PCA) is a process that helps us visualize the variables and their observations in a low-dimensional space. This tool is very interesting especially when a dataset contains more than two variables, because it allows us to compare the relationship of our variables at the same time without having any multidimensional visualization. PCA works by finding a low-dimensional space of our dataset that includes the highest information variety. Each one of these dimensions is a linear mixture of our variables.

Finally, we are going to examine the distribution of our dependent variable (quality) in order to see if it's a normal one. Normality (or normal distribution) has to do with the symmetrical distribution of our variable's observations around its mean in a way that shapes a bell-curved histogram. In statistics normality is important because it shows that our dataset (and its observations) is modeled in a good way that could result in reliable and credible statistical results.

Red wines

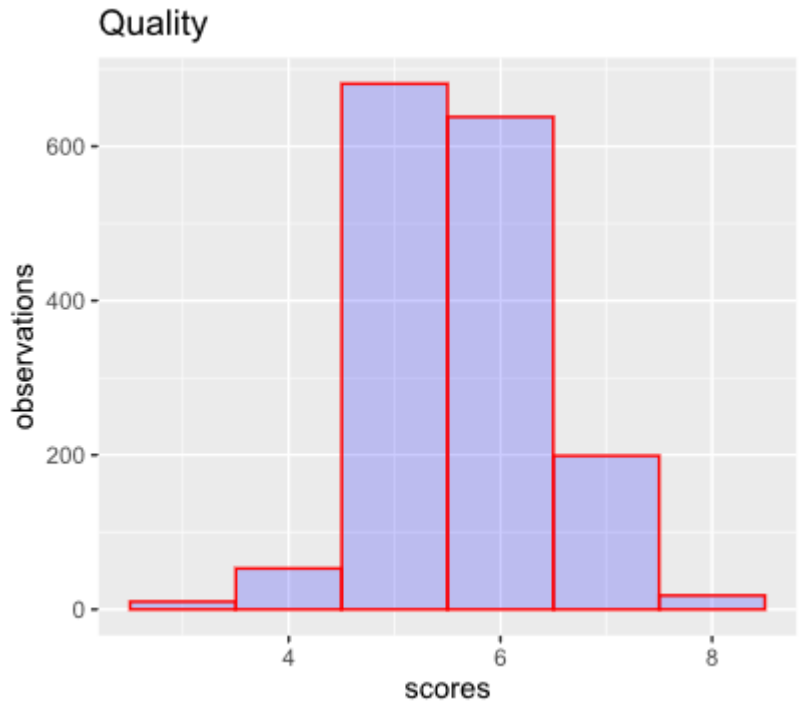
- Secondary Data
- Dataset's URL: http://www3.dsi.uminho.pt/pcortez/wine/
- Variable Analysis:

VARIABLES	MEANING	MIN VALUE	MAX VALUE
QUALITY (integer,discrete,dependent)	Indicates the quality level of the wine	3	8
FIXED ACIDITY (numeric,continuous,independent)	Indicates the fixed acidity level of the wine	4.6	15.9
VOLATILE ACIDITY (numeric,continuous,independent)	Indicates the volatile acidity level of the wine	0.12	1.58
CITRIC ACID (numeric,continuous,independent)	Indicates the citric acid level of the wine	0	1
RESIDUAL SUGAR (numeric,continuous,independent)	Indicates the residual sugar level of the wine	0.900	15.500
CHLORIDES (numeric,continuous,independent)	Indicates the chlorides level of the wine	0.012	0.611
FREE SULFUR DIOXIDE (numeric,continuous,independent)	Indicates the free sulfur dioxide level of the wine	1	72
TOTAL SULFUR DIOXIDE (numeric,continuous,independent)	Indicates the total sulfur dioxide level of the wine	6	289
DENSITY (numeric,continuous,independent)	Indicates the density of the wine	0.9901	1.0037
PH (numeric,continuous,independent)	Indicates the PH level of the wine	2.740	4.010
SULPHATES (numeric,continuous,independent)	Indicates the sulphates level of the wine	0.33	2
ALCOHOL(numeric) (continuous)	Indicates the alcohol level of the wine	8.4	14.9

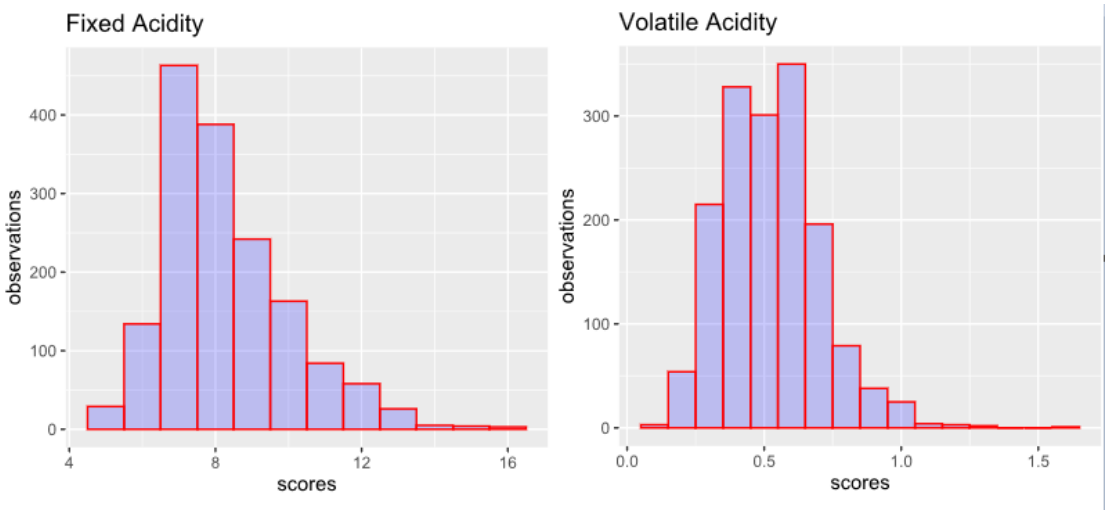
- Dataset's Structure

OBSERVATIONS	VARIABLES
1599	12

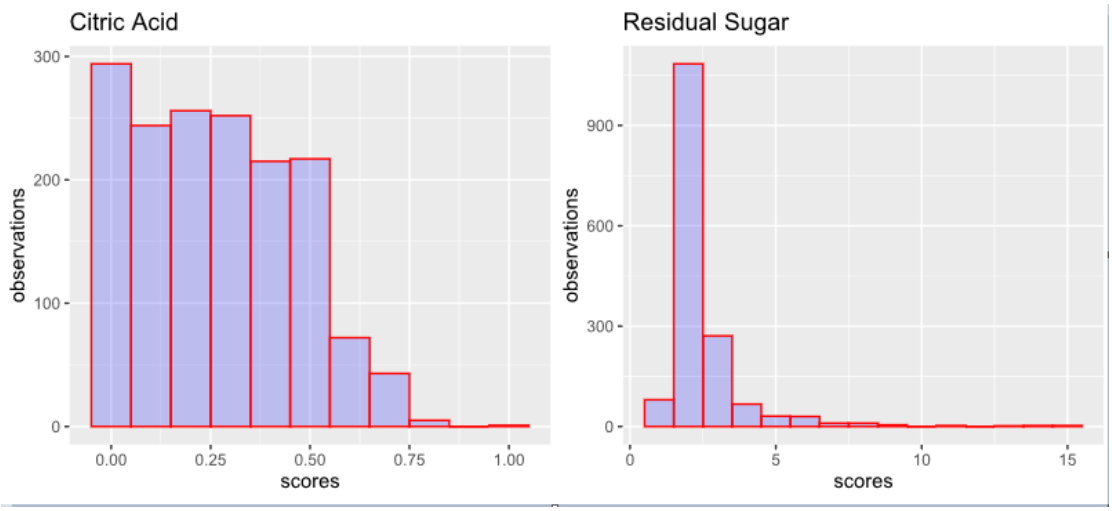
- Graphical Analysis(Histograms)



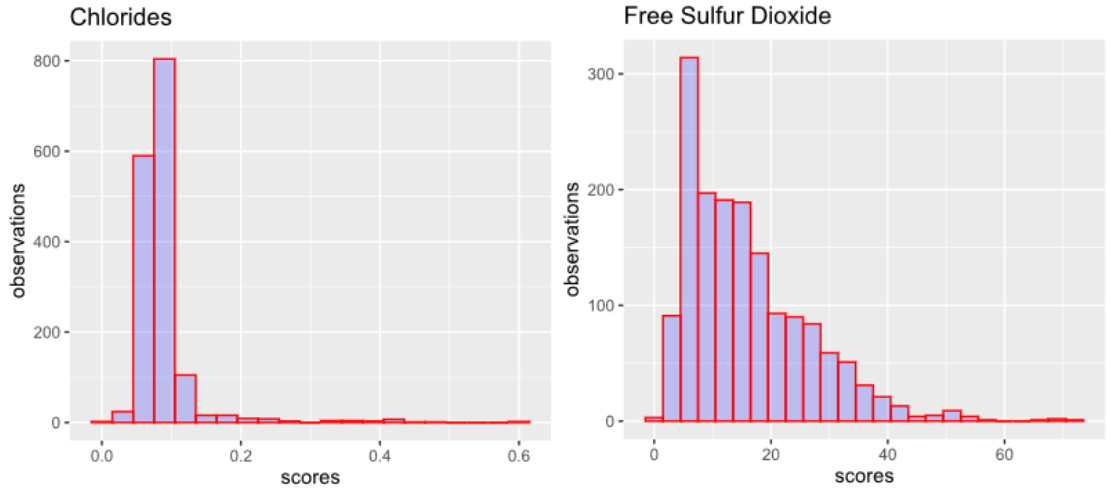
Comment:
As we can see, the Quality histogram seems to be symmetric with its peak being in the middle of its shape.



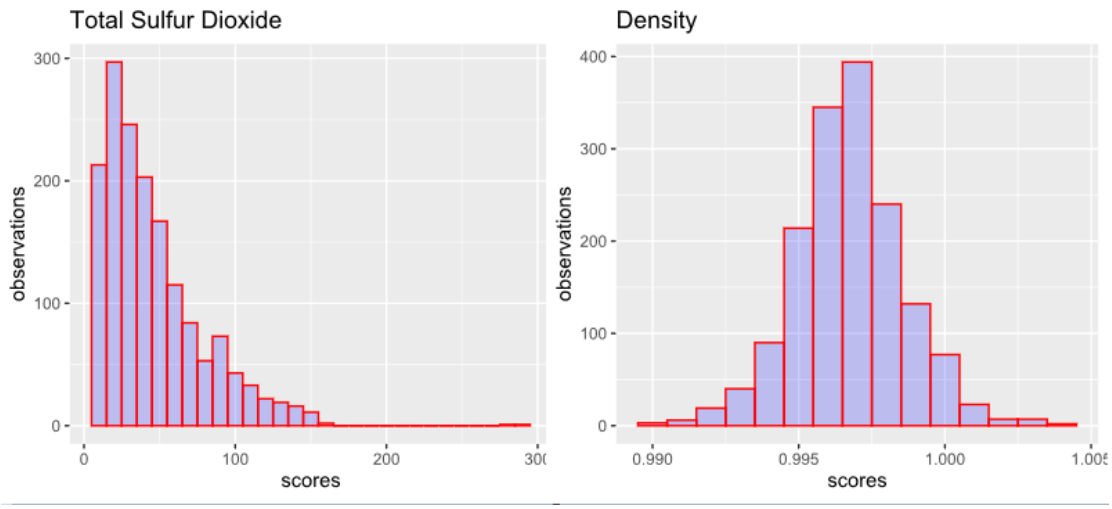
Comment:
As we can see both Fixed Acidity and Volatile Acidity histograms are right-skewed. Fixed Acidity has its peak in the left side of its shape contrary to Volatile Acidity which has its peak in the middle.



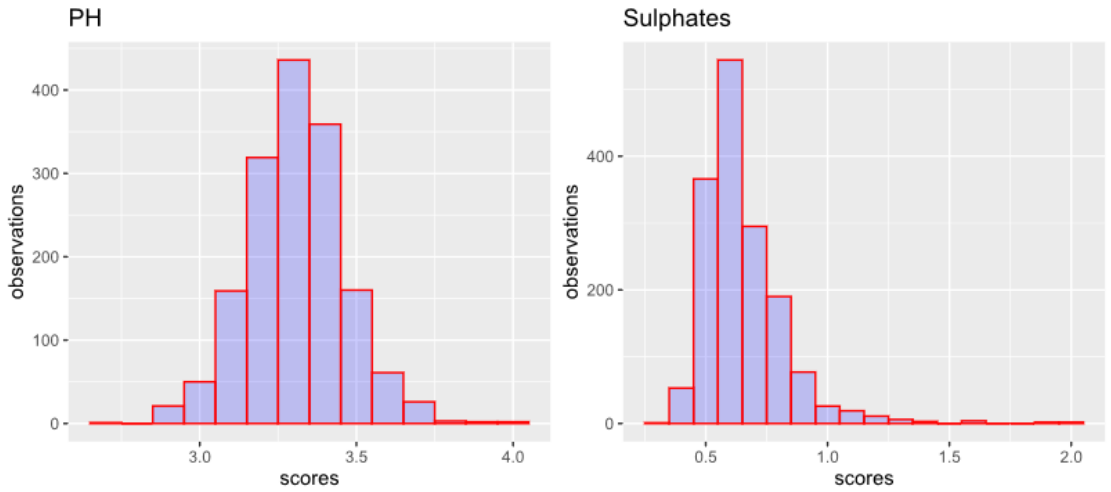
Comment:
As we can see both Citric Acid and Residual Sugar histograms are right-skewed with their peaks being in the left side of their shape.



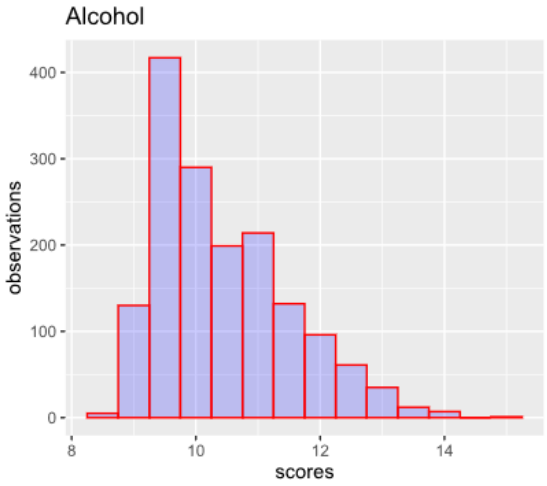
Comment:
As we can see both Chlorides and Free Sulfur Dioxide are right-skewed with their peaks being in the left side of their shape.



Comment:
As we can see, the Total Sulfur Dioxide histogram is right-skewed with its peak being in the left side of its shape. From the other hand the Density histogram seems to be a symmetric one with its peak being exactly in the middle of its shape.



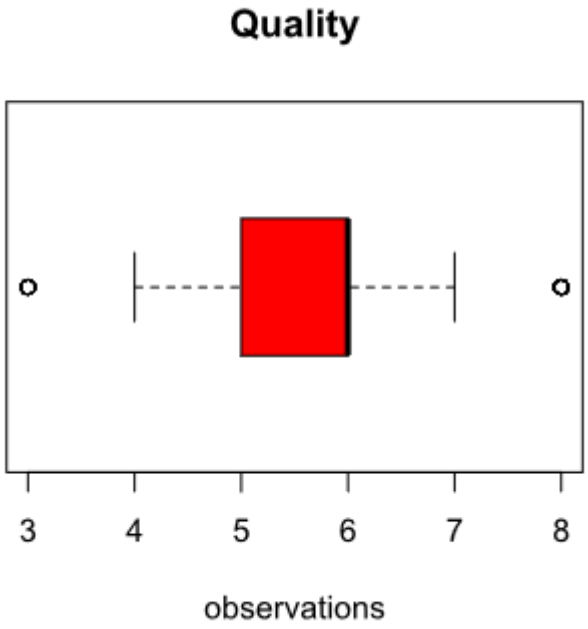
Comment:
As we can see, the PH histogram seems to be a symmetric one with its peak being exactly in the middle of its shape, contrary to the Sulphates histogram which is right-skewed and it has its peak in the left side of its shape.



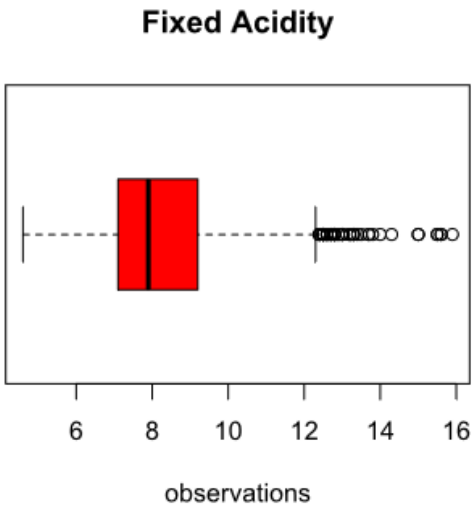
Comment:

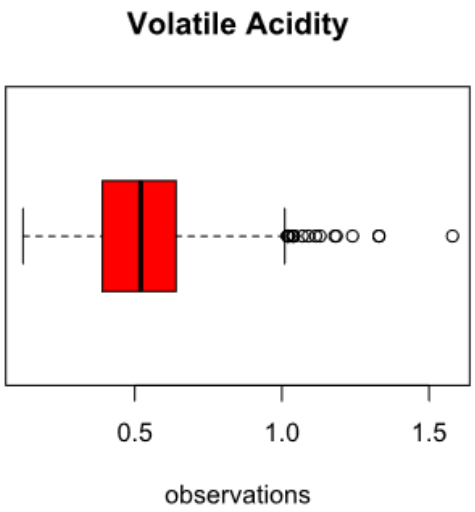
Last but not least, as we can see, the Alcohol histogram is right-skewed with its peak being in the left side of its shape.

- Graphical Analysis (Boxplots)

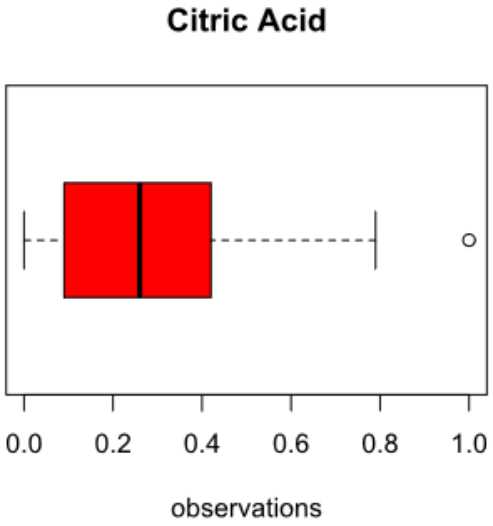


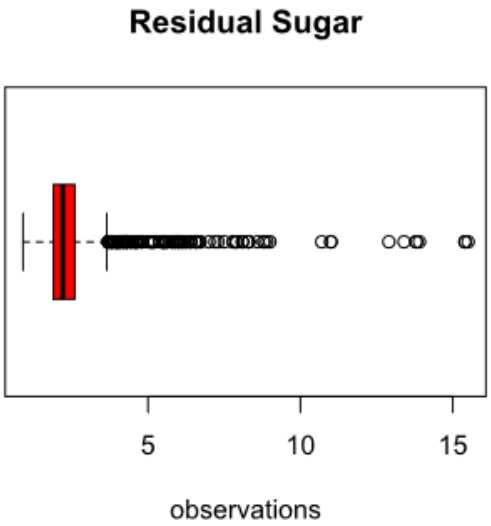
VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
QUALITY	28	1.7



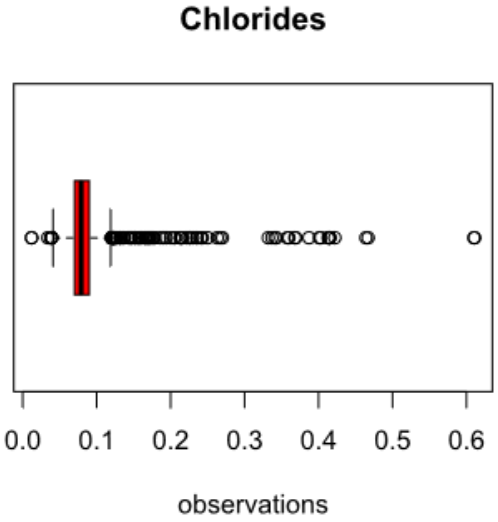


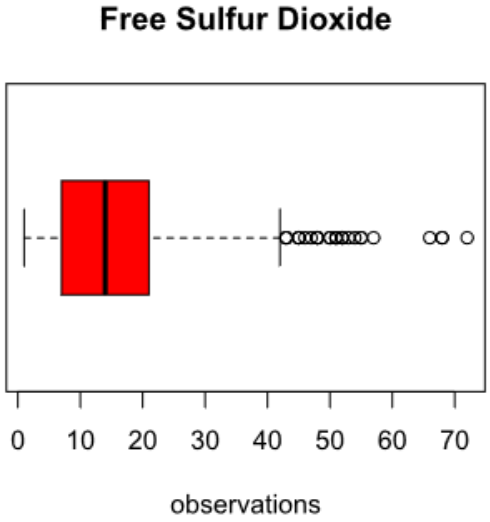
VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
FIXED ACIDITY	50	3
VOLATILE ACIDITY	20	1.2



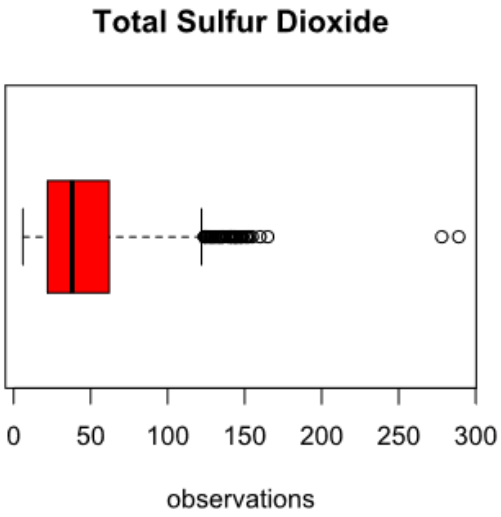


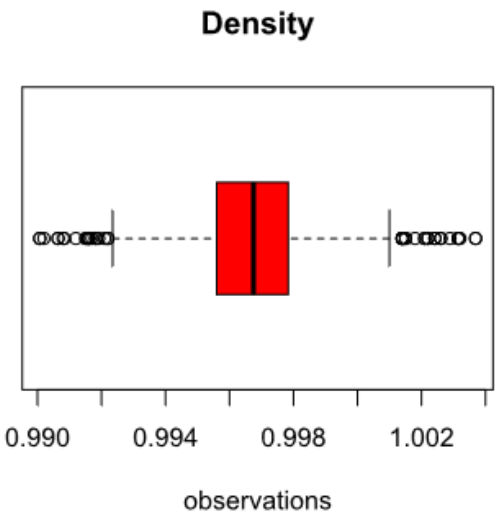
VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
CITRIC ACID	1	0
RESIDUAL SUGAR	156	9



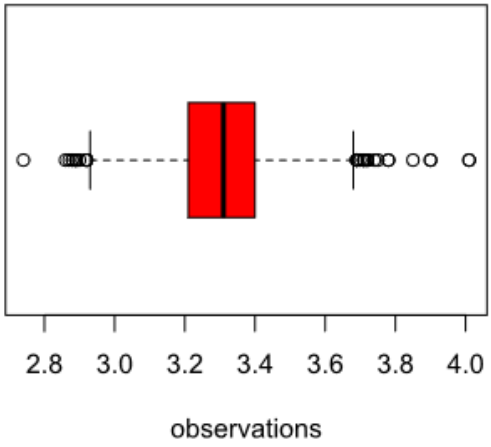


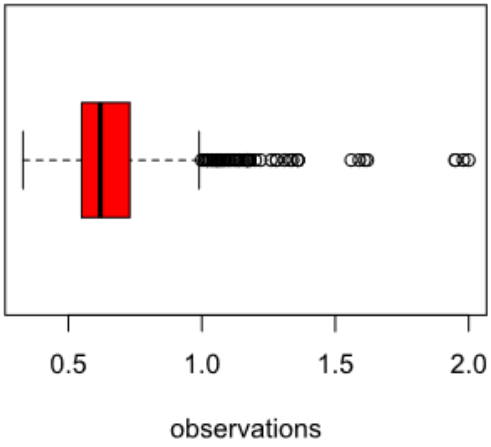
VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
CHLORIDES	113	7
FREE SULFUR DIOXIDE	30	1.8



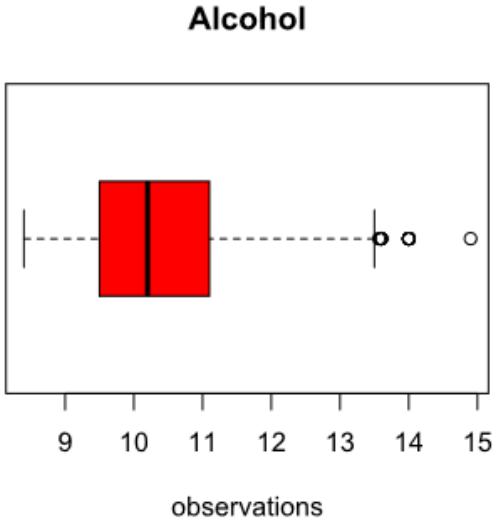


VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
TOTAL SULFUR DIOXIDE	56	3.6
DENSITY	46	2.8





VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
PH	36	2.2
SULPHATES	60	3.7

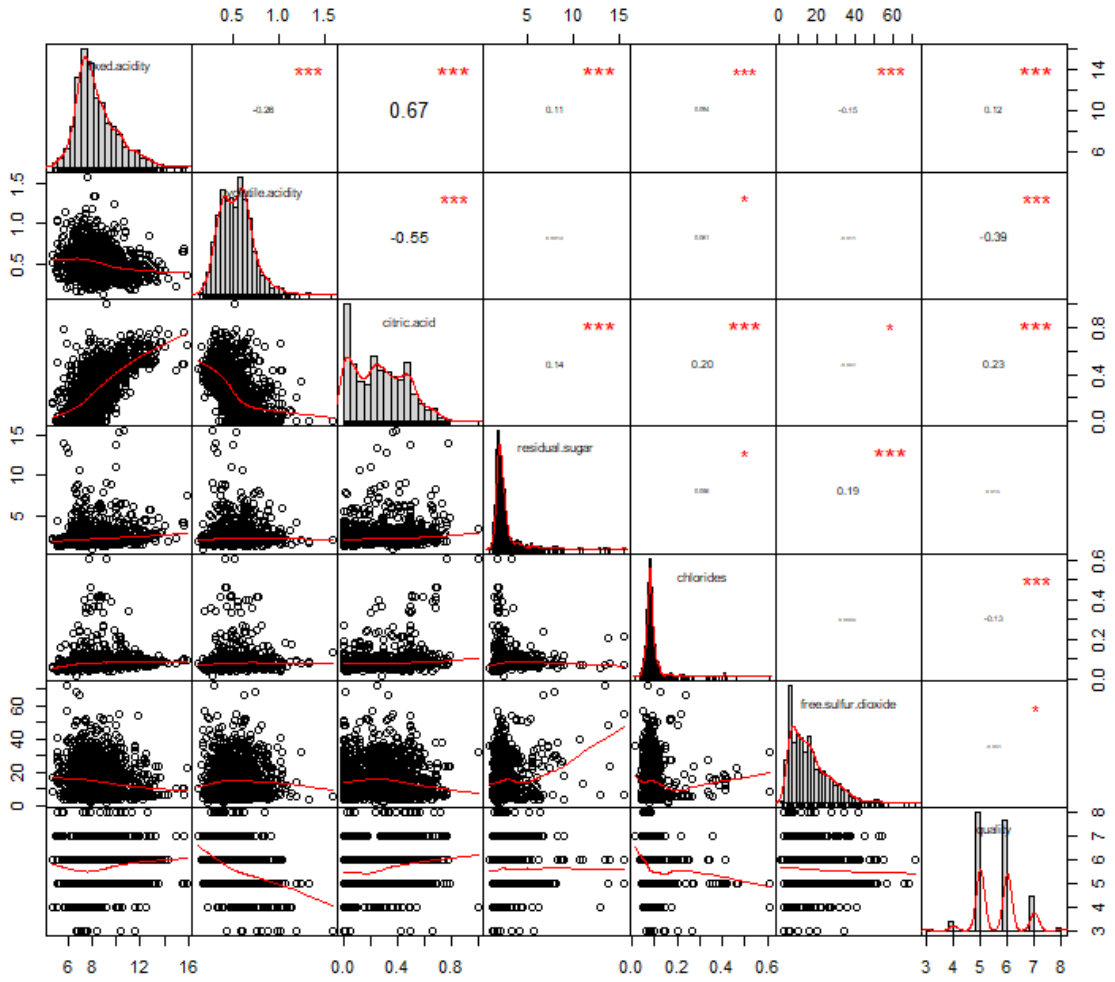


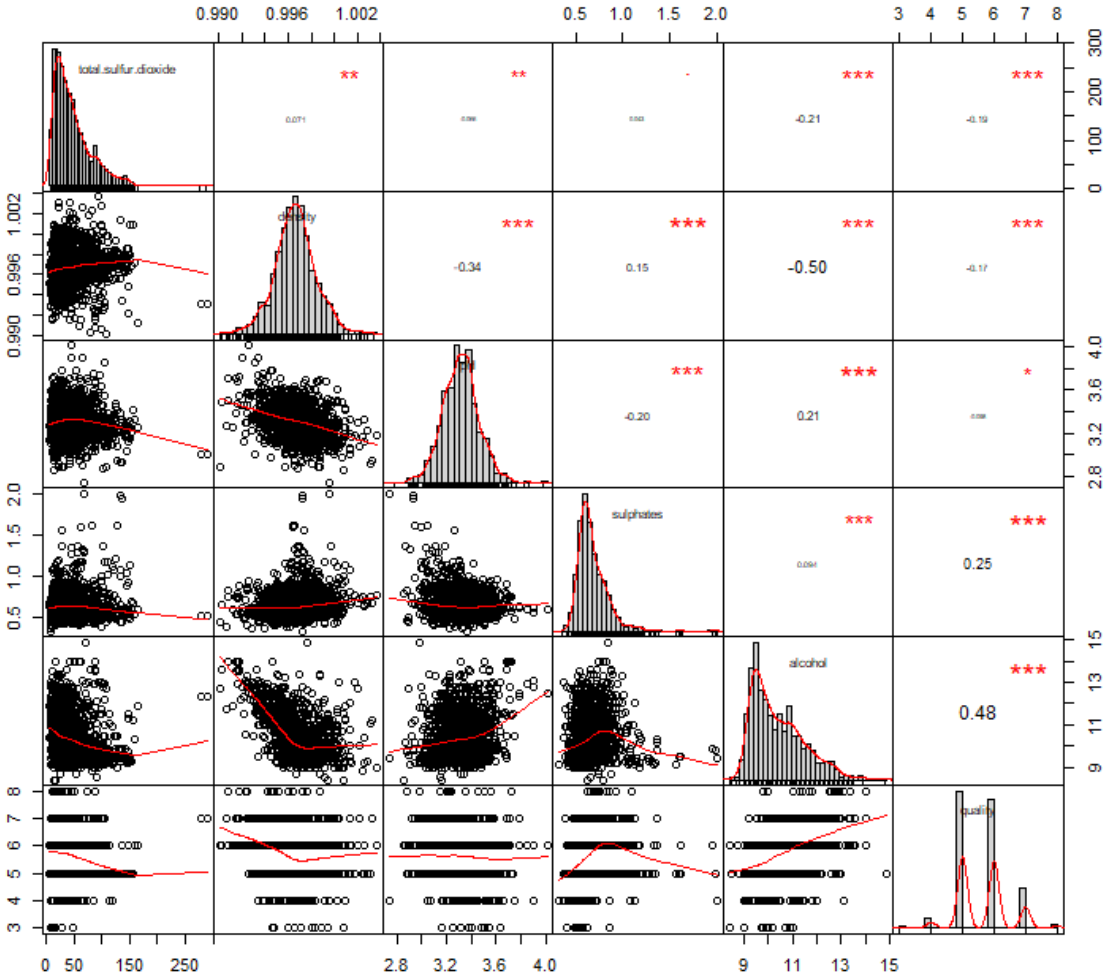
VARIABLES	OUTLIERS (amount)	OUTLIERS (percentage)
ALCOHOL	14	0.8

- Variance, Standard Deviation, Standard Error

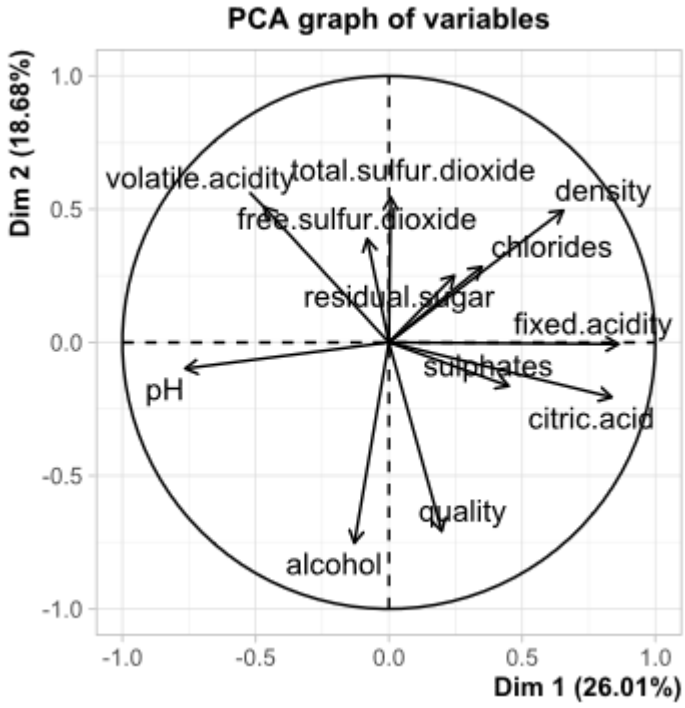
	VAR	SD	SE
FIXED ACIDITY	3.031	1.741	0.043
VOLATILE ACIDITY	0.032	0.179	0.004
CITRIC ACID	0.037	0.194	0.004
RESIDUAL SUGAR	1.987	1.409	0.035
CHLORIDES	0.002	0.047	0.001
FREE SULFUR DIOXIDE	109.414	10.460	0.261
TOTAL SULFUR DIOXIDE	1082.102	32.895	0.822
DENSITY	3.562029e-06	0.001	4.71981e-05
PH	0.023	0.154	0.003
SULPHATES	0.028	0.169	0.004
ALCOHOL	1.135	1.065	0.026
QUALITY	0.652	0.807	0.020

- Correlation




total.sulfur.dioxide
density
sulphates
alcohol
quality
0.990
0.996
1.002
0.5
1.0
1.5
2.0
3
4
5
6
7
8
0
100
200
300
2.8
3.2
3.6
4.0
9
11
13
15
0
50
150
250
-0.21
-0.50
0.25
0.48


fixed.acidity
-0.26
volatile.acidity
0.67
-0.55
citric.acid
0.11
0.14
esidual.sugar
0.09
0.06
0.2
0.06
chlorides
-0.15
0.19
e.sulfur.dioxide
-0.11
0.08
0.2
0.67
l.sulfur.dioxide
0.67
0.36
0.36
0.2
density
-0.68
0.23
-0.54
-0.27
pH
0.18
-0.26
0.31
0.37
sulphates
-0.2
0.11
-0.5
0.21
0.09
alcohol
0.12
-0.39
0.23
-0.13
0.25
0.48
quality
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


PCA graph of variables
Dim 2 (18.68%)
Dim 1 (26.01%)
1.0
0.5
0.0
-0.5
-1.0
volatile.acidity
total.sulfur.dioxide
density
free.sulfur.dioxide
chlorides
residual.sugar
fixed.acidity
sulphates
pH
citric.acid
quality
alcohol

Comment:
From the four diagrams (correlation plots and PCA) and the table above we can see that there is negative and positive correlation between our variables. Our dependent variable (quality) has the strongest positive correlation with alcohol (0.476) and the strongest negative correlation with volatile acidity (-0.391). Furthermore, the dependent variable seems to have positive bondages with 4 more independent variables (fixed acidity, citric acid, residual sugar, sulphates) and negative bondages with 5 more independent variables (chlorides, free sulfur dioxide, total sulfur dioxide, density, ph).

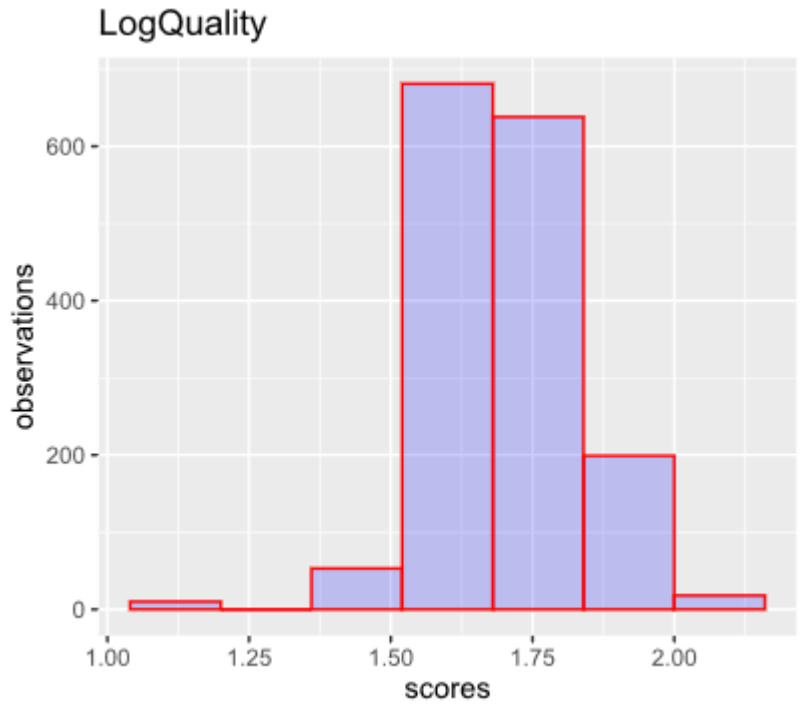
- We Check For Normality In The Dependent Variable

TEST	**VALUES (p.value)**
SHAPIRO-WILK	<2.2e-16
JARQUE-BERA	0.002
ANDERSON-DARLING	<2.2e-16
CRAMER-VON MISES	7.37e-10
KOLMOGOROV-SMIRNOV	<2.2e-16

Comment:
As we can see from the Shapiro, Jarque-Bera, Anderson-Darling, Cramer-von Mises and Kolmogorov-Smirnov tests, p.values are smaller than 0.05 **(p.values < 0.05 means that there is no normality in our distribution).** These two results indicate that our dependent variable (quality) is not normally distributed.

- We Transform The Type Of The Dependent Variable And Then We Check Again For Normality

TYPE OF DEP.VARIABLE	**VALUES**
NO-TRANSFORMATION	59.37
BOX-COX	59.271
LOG_B(X+A)	59.180
SQRT(X+A)	59.271
EXP(X)	59.189
ARCSINH(X)	59.187
YEO-JOHNSON	59.271
ORDERNORM	59.265



TEST	VALUES (p.value)
SHAPIRO-WILK	<2.2e-16
JARQUE-BERA	<2.2e-16
ANDERSON-DARLING	<2.2e-16
CRAMER-VON MISES	7.37e-10
KOLMOGOROV-SMIRNOV	<2.2e-16

Comment:
As we can see from the table above, after a transformation in the type of the dependent variable (in seven different forms: Box-Cox, Log, sqrt, exp, arcsinh, Yeo-Johnson, ordernorm), we notice that there is no significant change in the levels of our normality. Nevertheless, there is a tiny improvement in every transformed type contrary to our early stage and especially in the log transformation of our explanatory variable, because a lower index indicates a more normal distribution in our data. Furthermore, we transformed our dependent variable into a log type and we checked again using a histogram and the five normality tests (Shapiro, Jarque-Bera, Anderson-Darling, Cramer-von Mises and Kolmogorov-Smirnov), that gave us the same negative (considering normality) results.

Chapter 2

Multiple linear regression

Theoretical background

According to Auckland University lecturer Gareth James and his book on statistics (James et al.) and University of Macedonia professors Ioannides Dimitrios & Athanasiadis Ioannis and their book on R and statistics (Athanasiadis.I and Ioannides.D, 2017) linear regression is a type of regression that is used in order to predict quantitative values and results (e.g. salaries-wages, income, prices etc.). It is the simplest form of regression and ties the relationship between one dependent variable and one (or more) independent variables. This type of regression can be found in many statistical textbooks all over the world and it's the oldest and most used tool for predictions. Linear regression is divided in two different forms (Simple Linear Regression & Multiple Linear Regression) depended on the number of the independent-explanatory variables of the model.

Simple linear regression consists of two variables, the dependent variable (Y) and the explanatory variable (X) and in mathematics can be described as $\mathbf{Y \approx \beta 0 + \beta 1 X}$ ($\mathbf{\beta 0}$ and $\mathbf{\beta 1}$ represent the model's intercept and coefficient, respectively). Just like its advanced type (Multiple Linear Regression), simple regression contains two very important statistic metrics (R-squared and RSE). Basically, RSE is an estimation of the average amount of errors that diverge from the regression line and calculates the model's lack of fit to its data, while R-squared measures the same thing in an alternative way which is independent of the dependent variable (Y). This difference gives R-squared an advantage over RSE, since it is always between zero and one (unlike RSE), but both metrics provide us information about the linear regression fit to the data, which is very important for our model's prediction accuracy.

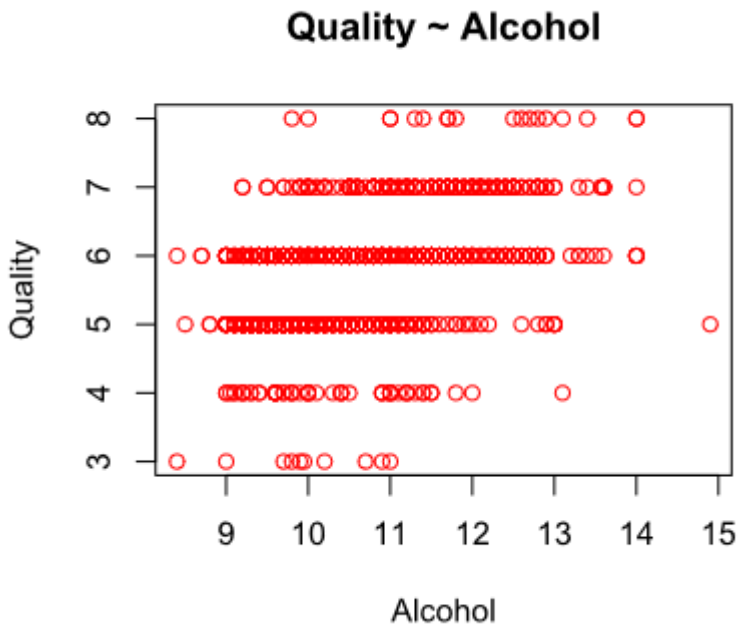
Moving on, if we add more than one independent variable in our simple linear regression we will end up having its advanced form, the multiple linear regression. MLR is one of the most well-known regression types around the world and mathematically can be described as $\mathbf{Y = \beta 0 + \beta 1 X1 + \beta 2 X2 + \cdots + \beta pXp + \varepsilon}$. However, in multiple regression one of the main issues has to do with the decision on important variables for our final model. As we know from our experience, a dataset is possible to contain many variables that aren't statistically significant in order to be part of our regression model. For that reason, there are three different approaches that have been developed in order to conduct a variable selection and are known as forward, backward and stepwise-mixed selections. Each one of the previous methods can be done with many different ways (classic, regsubset or step functions) depending on the criterion that's been used (AIC or BIC) in order to distinct our variables in significant and insignificant and their difference lies on the order in which they delve the dataset's components.

After having performed the variable selection it is time to make predictions using our final regression model, but before that it is important to examine if our regression model fulfills the regression assumptions. The assumptions' number vary from textbook to textbook and generally are four for simple linear regression and five for multiple linear regression. They examine our regression model in terms of linearity, homoscedasticity, independence of residuals, normality of residuals and multicollinearity. In case that our final model doesn't meet every single assumption, it is possible that could result in erroneous predictions and unreliable conclusions ("Assumptions of Regression Analysis, Plots & Solutions.", 2016) ("What Are the Four Assumptions of Linear Regression? – Gaurav Bansal.").

Finally, in order to complete a linear regression analysis, we have to evaluate its prediction performance. This sector has to be done through a variety of prediction metrics (RMSE, MAE, MAPE, MSE, Min-Max Accuracy) that are responsible for telling us the accuracy and the prediction error possibility of our model. These metrics may differ depending on the way that our dataset has been separated in training and testing set. Because of that, in the following section we are going to conduct two different training-testing data separations: the classic one which divides the whole dataset in two parts (70% on training set and30% on testing set) and the second one (10 fold cross validation) which separates the data observations in ten different folders, using one of them as a testing set and the remaining nine as a training set.

Red wines

- We Build A Simple Regression Between Quality And Alcohol



Comment:
The analysis shows, that a simple regression between our dependent variable (Quality) and it's most correlated independent variable (Alcohol) would resulted in this regression: Quality= 1.874 + 0.360*Alcohol. Furthermore the R-squared (multiple and adjusted) would be around 22% (0.2267 and 0.2263) and that shows a low data concentration levels around our fitted regression line. Last but not least, due to the fact that our dependent variable is an integer and not a continuous one, the plot of these two variables would have its observations in a horizontal distribution like the one above us.

- We Check The Assumptions Of The Simple Regression

1. Linearity

Comment:
As we notice from our simple regression residuals plot, our regression is a linear one, because of the linear allocation of the residuals. This criterion is important, due to the fact that without this both coefficients and standard errors in our output would be unreliable.

2. No Existence Of Heteroscedasticity In Our Model

TEST	**VALUES (p.value)**
STUDENTIZED BREUSCH-PAGAN	6.454e-08

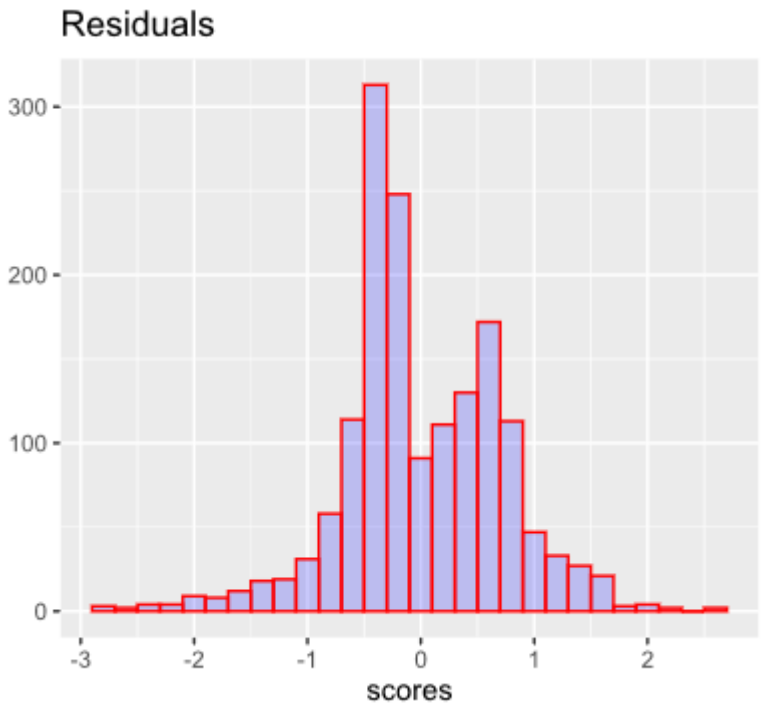
Comment:
From the Breusch-Pagan test we can clearly see that there is no homoscedasticity in our simple linear regression (**because the p.value is smaller than 0.05**). This means that in our linear regression line we have a different scatter (different variance for all values of the independent variable) and so our standard errors are wrong.

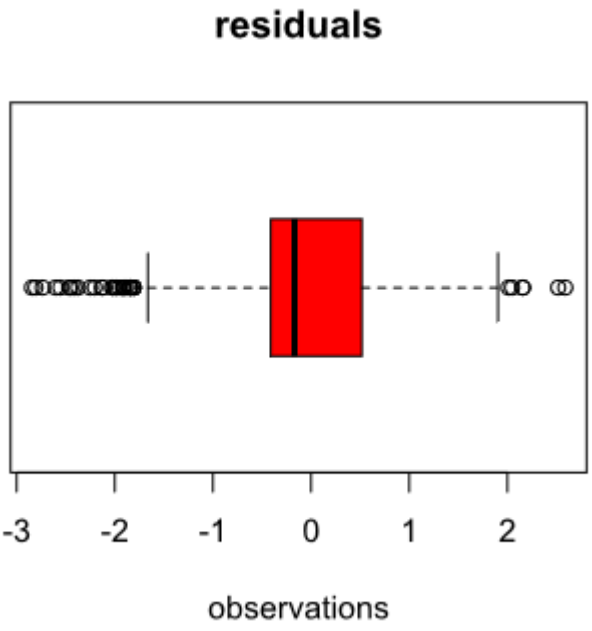
3. No Correlation Existence Between The Residuals And The Independent Variable

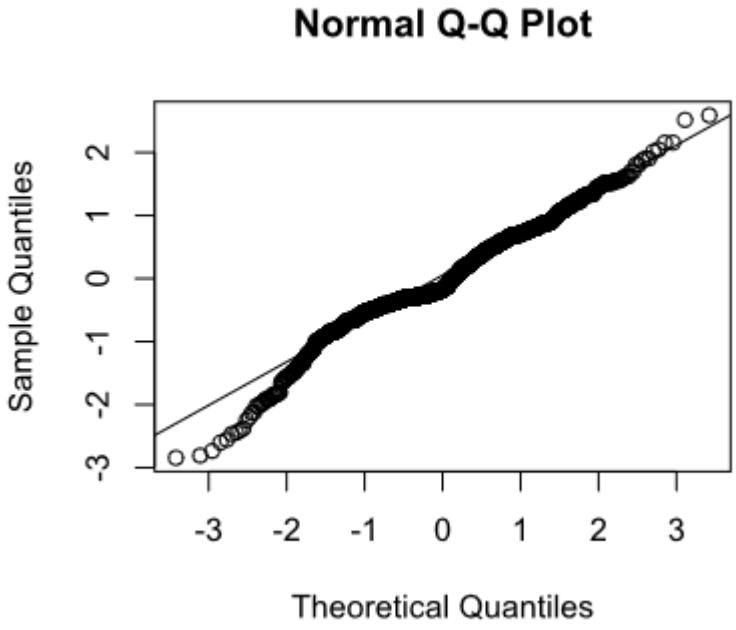
TEST	**VALUES (p.value)**
PEARSON'S PRODUCT-MOMENT	1

Comment:
Through a Pearson's product-moment correlation test we notice that p.value is high (bigger than 0.05) so there isn't correlation between the residuals and the independent variable (Alcohol), which is good, because it shows that our residuals are randomly distributed and aren't explained by our explanatory variable.

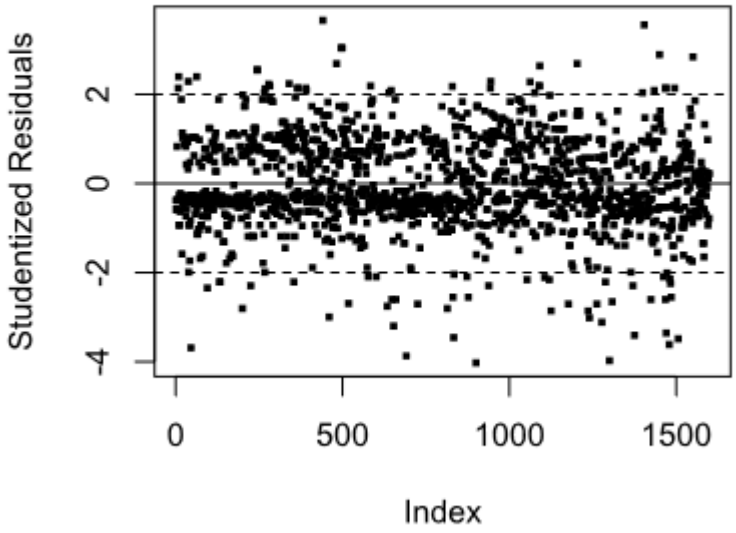
4. Normality Of Residuals

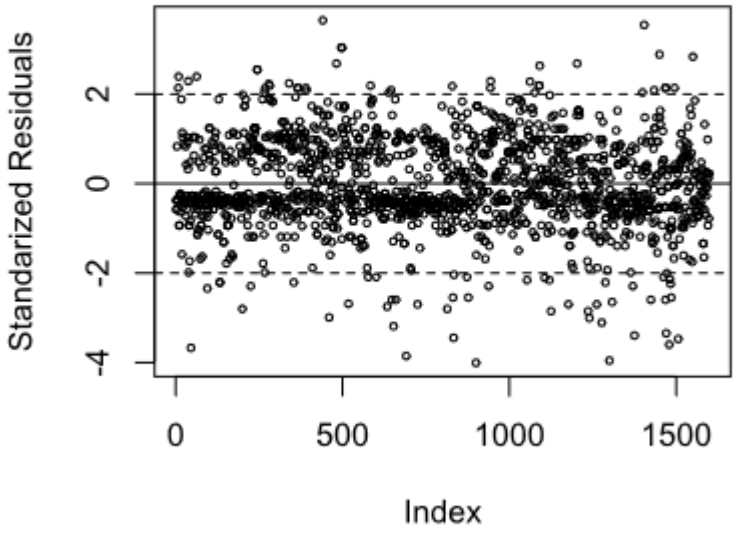






TEST	**VALUES (p.value)**
SHAPIRO-WILK	8.865e-16
JARQUE-BERA	<2.2e-16
ANDERSON-DARLING	<2.2e-16
CRAMER-VON MISES	7.37e-10
KOLMOGOROV-SMIRNOV	9.992e-15





Comment:
As we can see from the Shapiro, Jarque-Bera, Anderson-Darling, Cramer-Von Mises and Kolmogorov-Smirnov tests, the p.values are smaller than 0.05 **(p.values $<$ 0.05 means that there is no normality in our distribution)**. Nevertheless, the residuals' boxplot indicates only 36 outliers out of 1599 observations (2.2%) and so this criterion shows the exact opposite result than the previous ones

indicating the existence of normality in our residuals. Also, the remaining 4 plots (Histogram, Q-Q norm plot, Studentized Residuals plot and Standardized Residuals plot) seems to show a normal distribution in our residuals, due to the fact that there is a symmetrical distribution of observations around the mean and 97,8% (36 outliers out of 1599 observations or 2.2%) of observations in the interval and there is a close distribution in the normal quantiles (Q-Q norm).

5. No Autocorrelation

TEST	**VALUES (p.value)**
DURBIN-WATSON	0

Comment:
Autocorrelation is considered as one of the regression assumptions in many cases. Although, in our research the dataset that we use in order to conduct the statistical analysis is cross-sectional and we do not use time series. This small difference has as a result the insignificance of this specific test (autocorrelation test) because autocorrelation is a state of time dependence an in our case the dataset does not include this kind of chronical dimension. Nevertheless, this durbin-watson test showed that there is autocorrelation in our simple regression's residuals (because p.value is smaller than 0.05 which indicates autocorrelation existence), but it shouldn't be considered as a reliable and valid result due to the aforementioned dataset characteristics (StackExchange, 2013).

- We Build The Multiple Linear Regression With Every Variable

Comment:
The analysis shows, that the intercept is b0=21.965. Furthermore b1=0.025, b2=-1.084, b3=-1.083, b4=0.016, b5=-1.874, b6=0.004, b7=-0.003, b8= -17.881, b9=-0.414, b10=0.916, b11=0.276. Also, R-squared (multiple and adjusted) is around 35% (0.3606 and 0.3561)and that shows low data concentration levels around our fitted regression line.
Quality= 21.965 + 0.025*FixedAcidity -1.084*VolatileAcidity – 0.183*CitricAcid + 0.016*ResidualSugar – 1.874*Chlorides + 0.004*FreeSulfurDioxide – 0.003*TotalSulfurDioxide - 17.881*Density – 0.414*PH + 0.916*Sulphates + 0.276*Alcohol.

- We Build The Final Regression

Comment:
Having conducted forward, backward and stepwise method through regsubset formula, we conclude that there are 6 statistical significant independent variables (each one of them has a p.value < 0.05) that we can keep in our final regression (Volatile Acidity, Chlorides, Total Sulfur Dioxide, PH, Sulphates and Alcohol). Furthermore we notice that the intercepts is b0= 4.296. Also b1=-1.038, b2=-2.002, b3=-0.002, b4=-0.435, b5=0.889, b6=0.291 and there is an R-squared (multiple and adjusted) around 35% (0.3572 and 0.3548) that shows low data concentration levels around our fitted regression line.
Quality=b0= 4.296 - 1.038*VolatileAcidity - 2.002*Chlorides - 0.002*TotalSulfurDioxide - 0.435*PH + 0.889*Sulphates + 0.291*Alcohol.

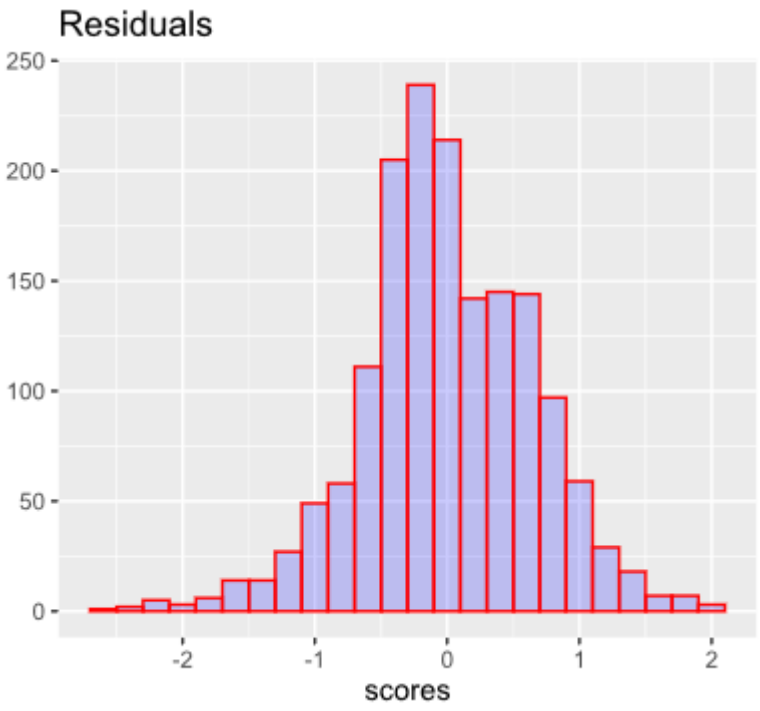
- We Check The Assumptions Of The Multiple Regression

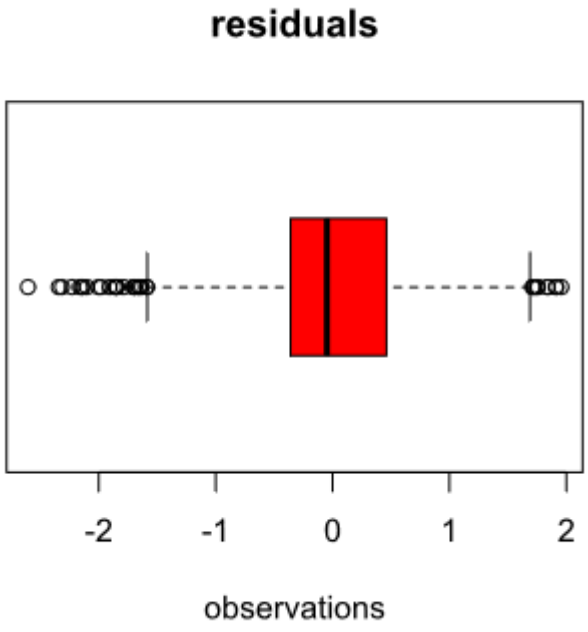
1. No Existence Of Multicollinearity

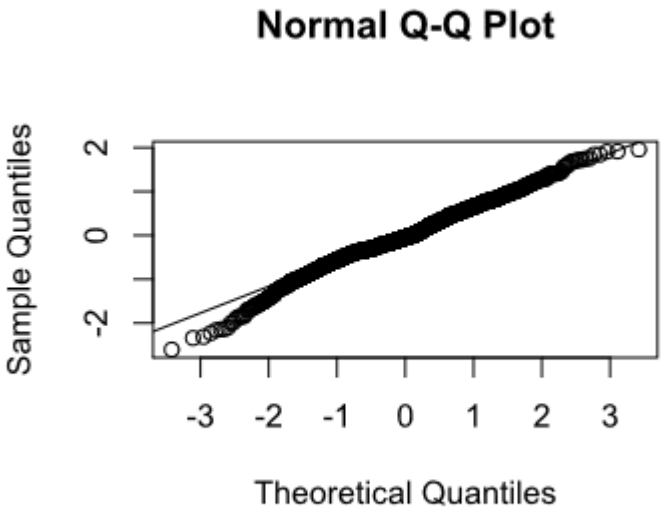
VARIABLES	**VIF VALUES**
VOLATILE.ACIDITY	1.227
CHLORIDES	1.332
TOTAL.SULFUR.DIOXIDE	1.053
PH	1.218
SULPHATES	1.321
ALCOHOL	1.218

Through a VIF test we notice, that every single independent variable has VIF values smaller than 5 **(VIF value $<$ 5 means that multicollinearity doesn't exist)**. These results indicate lack of multicollinearity among our independent variables which means thatevery independent variable in our model is unique and isn't already explained by other explanatory variables. This results give credibility in our model.

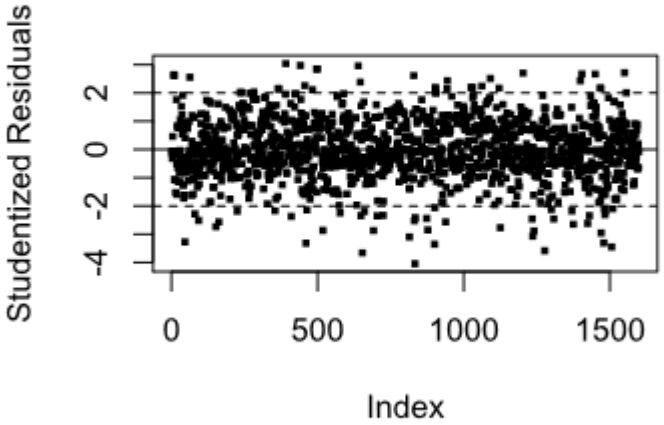
2. Normality Of Residuals

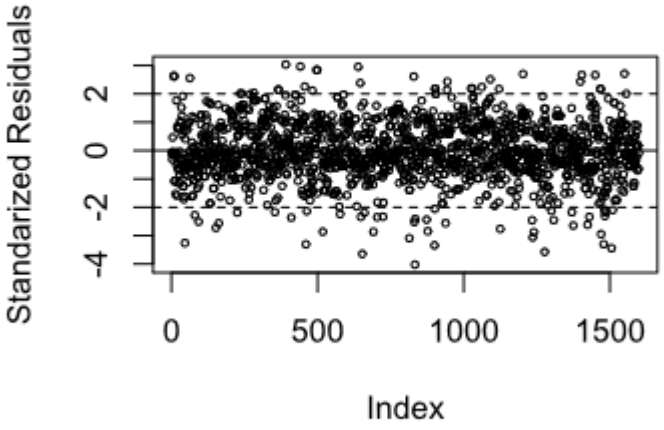






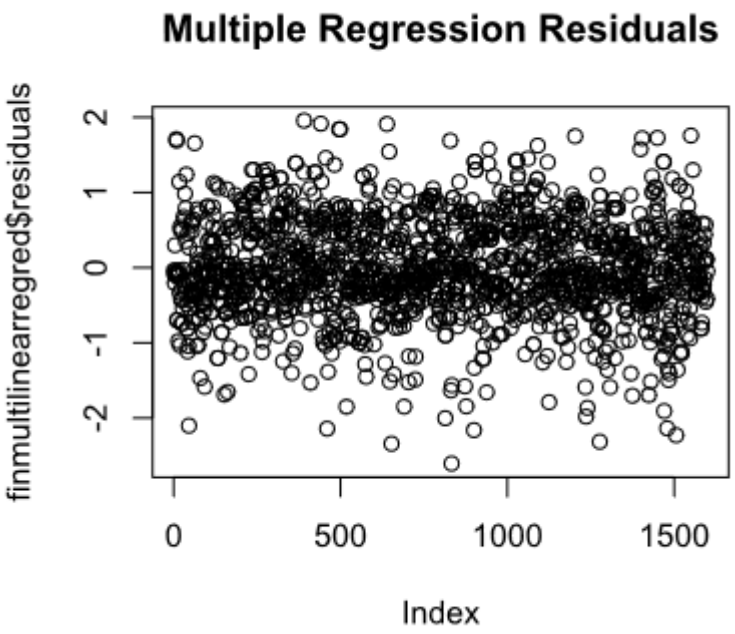
TEST	**VALUES (p.value)**
SHAPIRO-WILK	2.848e-08
JARQUE-BERA	<2.2e-16
ANDERSON-DARLING	6.035e-10
CRAMER-VON MISES	5.316e-08
KOLMOGOROV-SMIRNOV	0.001841





Comment:
As we can see from the Shapiro, Jarque-Bera, Anderson-Darling, Cramer-Von Mises and Kolmogorov-Smirnov tests, thep.values are smaller than 0.05**(p.values $<$ 0.05 means that there is no normality in our distribution)**. Nevertheless, the residuals' boxplot indicates only 34 outliers out of 1599 observations (2.1%) and so this criterion shows the exact opposite result than the previous ones indicating the existence of normality in our residuals. Also, the remaining 4 plots (Histogram, Q-Q norm plot, Studentized Residuals plot and Standardized Residuals plot) seems to show a normal distribution in our residuals, due to the fact that there is a symmetrical distribution of observations around the mean and 97,9% (34 outliers out of 1599 observations or 2.1%) of observations in the interval and there is a close distribution in the normal quantiles (Q-Q norm).

3. Linearity



Comment:
As we notice from our multiple regression residuals plot, our regression is a linear one, because of the linear allocation of the residuals. This criterion is important, due to the fact that without this both coefficients and standard errors in our output would be unreliable.

4. No Existence Of Heteroscedasticity In Our Model

TEST	**VALUES (p.value)**
STUDENTIZED BREUSCH-PAGAN	7.75e-10

Comment:
From the Breusch-Pagan test we can clearly see that there is no homoscedasticity in our multiple linear regression (**because the p.value is smaller than 0.05**). This means that in our linear regression line we have a different scatter (different variance for all values of the independent variables) and so our standard errors are wrong.

5. No Correlation Existence Between The Residuals And The Independent Variables

VARIABLES	PEARSON'S PRODUCT-MOMENT VALUES (p.value)
ALCOHOL	1
VOLATILE.ACIDITY	1
CHLORIDES	1
TOTAL.SULFUR.DIOXIDE	1
PH	1
SULPHATES	1

Comment:
Through six different Pearson's product-moment correlation tests we notice that p.value is high (bigger than 0.05) in every single case, so there isn't correlation between the residuals and the independent variables (alcohol, volatile acidity, sulphates, ph, chlorides and total sulfur dioxide), which is good, because it shows that our residuals are randomly distributed and aren't explained by our explanatory variables.

6. No Autocorrelation

TEST	**VALUES (p.value)**
DURBIN-WATSON	0

Comment:
As we already discussed on the section above, even though durbin-watson test shows autocorrelation in the residuals, we shouldn't rely on this result due to the fact that our dataset is cross-sectional.

- We Separate The Whole Dataset In Training And Testing Set (70-30%)

- Prediction Metrics

PREDICION METRICS	**VALUES**
RMSE	**0.902**
MAE	**0.721**
MAPE	**0.132**
MSE	**0.813**
MIN MAX ACCURACY	**0.883**

- 10 Folders Cross Validation

METRICS	RMSE	MAE
70-30 %	**0.902**	**0.721**
10 folders cross validation	**0.651**	**0.504**

Comment: After having conducted a 10 fold cross validation in our final regression model, we notice that two of our prediction metrics (RMSE and MAE) have different values than before. RMSE= 0.651 which is better than before and MAE= 0.504 which is also better. Both predictions show a better forecasting accuracy than before. In conclusion, the 10 fold cross validation seems to be a more reliable way of data separation than the 70-30 % formula.

Chapter 3

Ridge and lasso regressions

Theoretical background

According to Auckland University lecturer Gareth James and his book on statistics (James et al.) ridge and lasso regressions are part of the non-linear world and belong to the shrinkage method. This method tries to make the least squares fit to the data by shrinking the coefficients which ultimately ends up reducing the variance of our models.

Starting with ridge regression, it has many similarities with OLS. For example, both methods minimize RSS in order to end up with coefficient estimates that are part of a good fitted model. The only difference between these two regression methods, is that in this one we use a turning parameter (λ) which is part of the shrinkage penalty (λj$\sum$β^2j). This specific penalty affects our model's function and remains small if the coefficients are close to zero. Moreover, if the turning parameter is exactly zero then our coefficient estimations would be the same with these of the linear regression and as λgrows towards our model's coefficients will approach zero (except of the intercept that would be stable in any case, due to the fact that the penalty above does not have an impact on it).

Furthermore, except from the shrinkage penalty difference, ridge regression, as opposed to OLS, produces a variety of coefficients depending on the value o λ and that's why the right selection of our turning parameter is very important and significant in order to end up with a reliable ridge regression model with coefficient estimations that are close to reality. Last but not least, it is well known that ridge regression has two major advantages. First of all it is more flexible than least squares, due to the effect of the turning parameter (as λ increases, variance decreases and bias increase) and secondly, ridge regression has another advantage, this time over best subset selection method. Because of its form, ridge regression can fit only one model and so the model-fitting process can be done very fast.

The second most famous regression of the shrinkage method is lasso regression. This type of regression is very similar to ridge regression, but has a strong advantage over the second one. Unlike ridge, lasso regression does not include every independent variable in the final model, because in this case the shrinkage of the coefficients can go all the way through, until they are exactly equal to zero. So in lasso regression we have variable selection just like in OLS and its model can be construed more easily than ridge's, especially when the dataset includes some variables that are more important than others for our predictions. Moreover, just like in ridge regression the parameter λ is very important in order to get the right coefficients and its selection is being conducted through cross validation, but also in this case a bad λselection can result in a model with a wrong amount of variables that could give us false prediction conclusions.

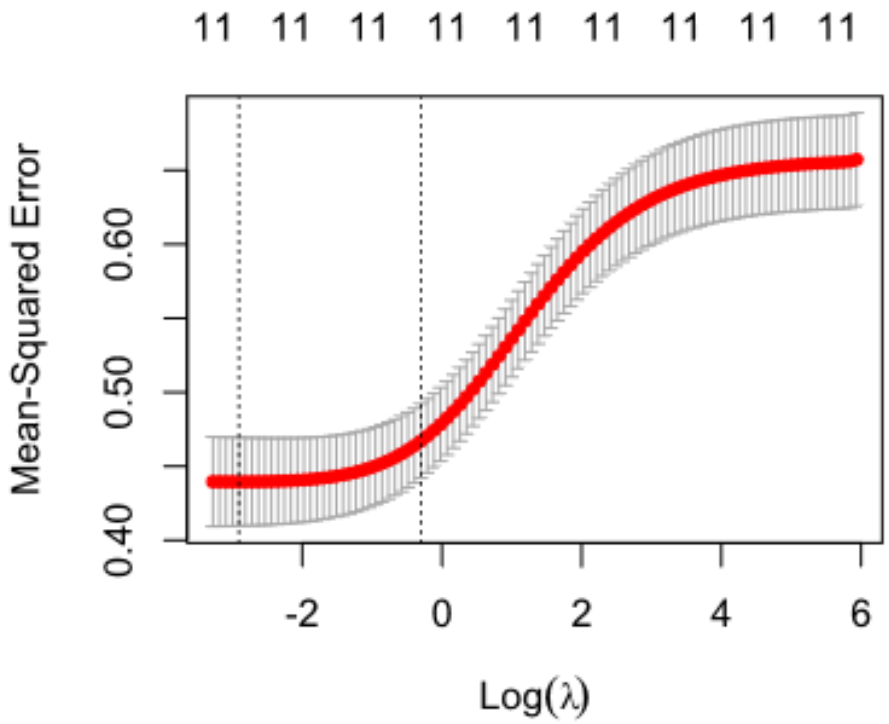
Last but not least, according to the book of Applied Multivariate Statistical Analysis (Härdle and Simar, 2019), lasso regression has three more little drawbacks (except of the one above) by itself. To be more specific, when the amount of the predictors (p) is bigger than the amount of the observations (n), lasso regression selects no more than n variables before it fills to capacity. Also, lasso method selects only one variable if there is a team of variables that have high correlation between each other and finally, if the predictors of the lasso model are correlated in a significant degree, the prediction accuracy of the lasso model is inferior to the one of the ridge model. In order to solve these trammels, Elastic net model was invented by Zou and Hastie. This method is made in such a way that can overcome the obstacles than are able to cause problems to the prediction accuracy of a lasso model, especially when the predictors are more than the data observations and there is high correlation between the variables.

Comparing ridge and lasso regressions we conclude that even if these two forms are quite similar to each other, lasso regression contains the major advantage of the variable selection, but except that difference there is another significant observation which has to do with the accuracy of these regression types. After having conducted an analysis of these two regression forms on two different datasets, the results showed that neither lasso nor ridge has the ultimate domination over the other. In fact, ridge regression showed to be a better prediction forecasting method when there are datasets in which the majority of the variables includes coefficients that are more or less of the same size and significance or in cases which high correlation exists between the predictors of the model, contrary to lasso regression which showed a better performance when it is applied on datasets in which a small part of the variables has fundamental coefficients and the remaining onehas coefficients that are close to zero.

Ridge regression

Red wines

- We Create The Right Functions For A Ridge Regression
- We Separate The Whole Dataset In Training And Testing Set
- We Find Out The Best λ For Our Model



Comment:
Through cross-validation, we notice that the best λfor our model is 0.05473216 and from the graphic above we can watch the full relation between λand MSE.

- We Build The Final Regression

Comment:
Our ridge regression model has concluded in eleven independent variable (ridge regression doesn't separate the variables in significant and insignificant). The intercept is b0=33.548 and the coefficients are b1=0.031, b2=-0.998, b3=-0.058, b4=0.019, b5=-1.790, b6=0.004, b7=-0.003, b8=-29.739, b9=-0.289, b10=0.875, b11=0.248.
Quality = 33.548+ 0.031*FixedAcidity - 0.998*VolatileAcidity - 0.058*CitricAcid + 0.019*ResidualSugar - 1.790*Chlorides + 0.004*FreeSulfurDioxide - 0.003*TotalSulfurDioxide - 29.739*Density - 0.289*PH + 0.875*Sulphates + 0.248*Alcohol.

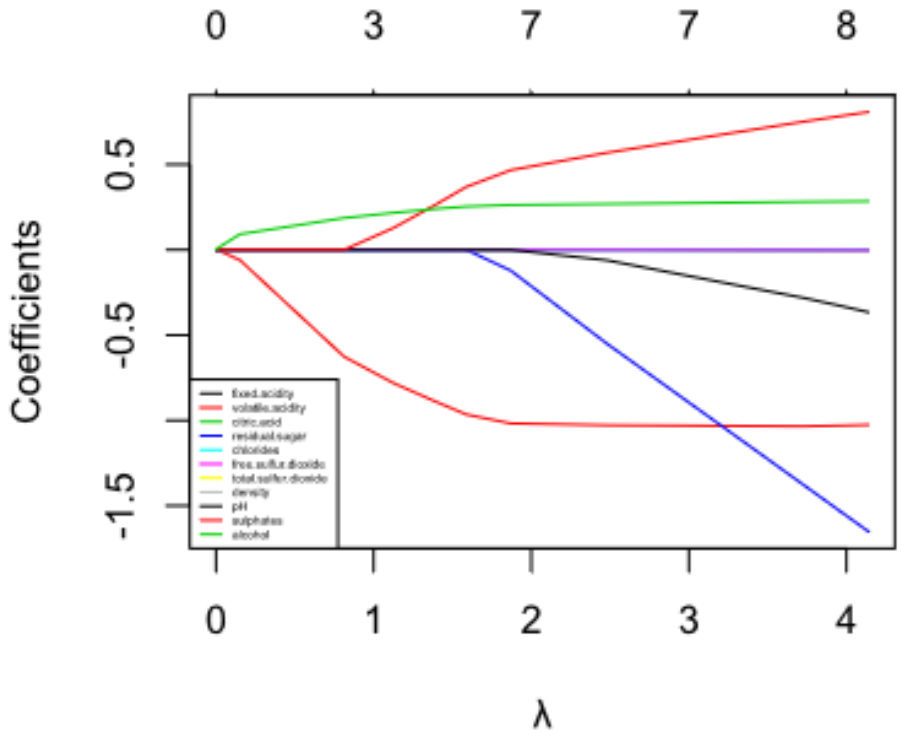
- Prediction Metrics

PREDICTION METRICS	VALUES
MSE	**0.638**
MAE	**0.499**
RMSE	**0.638**
MAPE	**0.091**

Lasso regression

Red wines

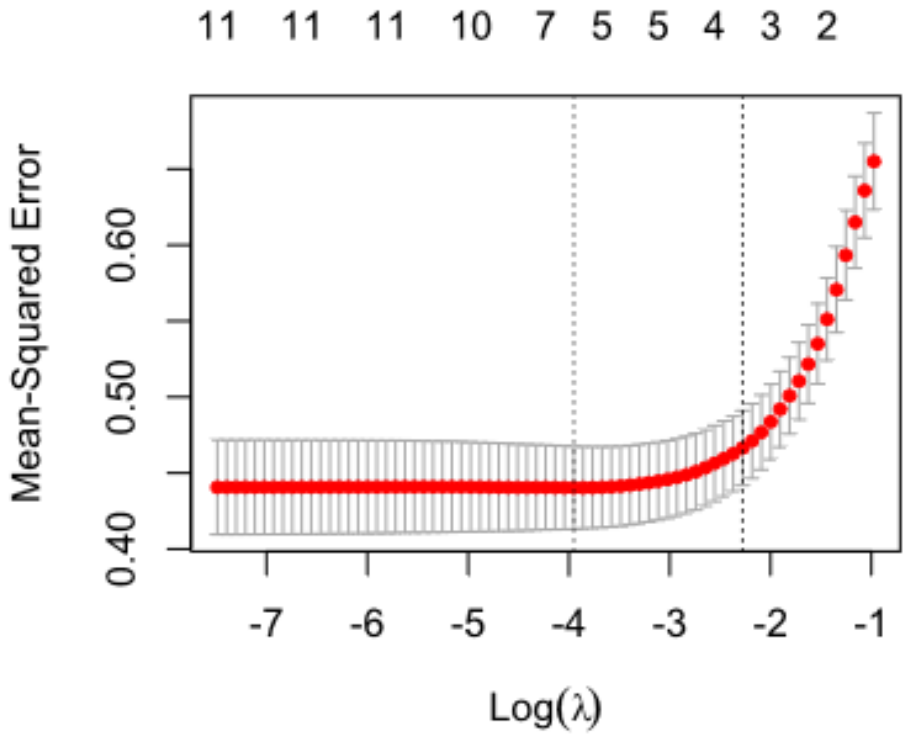
- We Create The Right Functions For A Lasso Regression And We Get The Lasso Plot



Comment:
The plot above shows the relation between λ and coefficients (only four variables have constantly zero coefficients: Free Sulfur Dioxide, Density, Residual Sugar and Citric Acid).

- We Separate The Whole Dataset In Training And Testing Set

- We Find Out The Best λ For Our Model



Comment:
Through cross-validation, we notice that the best λ for our model is 0.01921751 and from the graphic above we can watch the full relation between λ and MSE.

- We Build The Final Regression

Comment:
Our lasso regression model has concluded in six independent variable (lasso regression does separate the variables in significant and insignificant). The intercept is b0=3.852 and the coefficients are, b1=-1.034, b2=-1.315, b3=-0.002, b4=-0.265, b5=0.737, b6=0.280.
Quality = 3.852 - 1.034*VolatileAcidity - 1.315*Chlorides - 0.002*TotalSulfurDioxide - 0.265*PH + 0.737*Sulphates +0.280 *Alcohol.

- Prediction Metrics

PREDICTION METRICS	VALUES
MSE	**0.412**
MAE	**0.505**
RMSE	**0.642**
MAPE	**0.092**

Chapter 4

Generalized non-linear models

Theoretical background

1. Conforming to G. James (James et al.), the previous two linear models (ridge and lasso) tried to overcome the prediction limits of OLS by minimizing the model's coefficients. Nevertheless, there are still many more methods that tried to improve the primary prediction form of least squares, one of these methods is the polynomial regression.

In order to understand the polynomial method we have to have in mind the form of a simple linear regression. This simple model, that can be presented as $\mathbf{yi = \beta 0 + \beta 1 xi}$, is the theoretical framework of PR, due to the fact that the last one is just a d-degree representation of the previous model and it looks exactly like this : $\mathbf{yi = \beta 0 + \beta 1 xi + \beta 2 x\hat{}2i + \beta 3 x\hat{}3i + ... + \beta d x\hat{}di}$.In this new model the coefficients can be estimated using OLS, because beside the different form of the regression's predictors, everything else is still similar to our simple linear model.

Considering the degree in which we can raise our polynomial model, it shouldn't be more than 3 or 4, otherwise the regression's line form would end up being abnormal. Polynomial regression is a series of basic functions (bj(xi)=x^ji) that are applied on an explanatory variable and its relation with the linear model means that we can adopt every useful tool that we have used in least-squares model.

Even though polynomial regression can be conducted with the traditional-aforementioned series of basic functions, there are many other different function systems that can lead to the same or even better prediction results. One of these functions is called "Piecewise Polynomial" and contrary to the previous way, it provides us with an amount of low-degree polynomials that fit over different regions of our explanatory variable. This transformed category along with other function forms as "Constraints and Splines" is part of regression splines which is a flexible class of basic functions.This kind of polynomial procedure tends to give us better and more stable prediction results than before, because it doesn't need any high-degree exponents in order to produce models that fit well to the data.

2. Another significant type of a nonlinear model is the logistic regression. Taking into account G.James' book chapter about the logistic regression (James et al.), logistic regression tends to analyze the nonlinear relationship of a model that uses a categorical dependent variable and a variety of independent-explanatory variables. Furthermore, its existence is based on the possibility theory and most of the times, the dependent variable is separated in two major categories.

Comparing the logistic regression with the linear one, we notice that there is a series of differences between these two prediction models. First of all, the most noticeable characteristic of the logistic model is the fact that it uses a categorical dependent variable contrary to the linear regression which uses

only quantitative variables in order to predict the final results. Secondly, the linear models use OLS in order to predict their coefficient, while logistic regression predicts its numbers using the ratio test. This second difference has also as a result the acceptance of homoscedasticity from the side of the linear regression in opposition with the logistic model which tends to accept only the existence of heteroscedasticity in its residuals.

Historically, this type of regression was invented as an alternate model for the linear discriminant analysis and it's being used in a variety of science areas, like the social sciences sector and medicine. Moreover, logistic regression models that use more than one independent variable try to predict the frequency in which the two categories of the dependent variable appear, using the number 1 for the first category (with a possibility of success "p") and the number 0 for the second one (with a possibility of failure "1-p").

Finally, due to the nature of our logistic model's variables, there is no chance of having a normal distribution in our observations and homogeneity in the variances of our variables.

3. Moving on, our third regression of this specific category (generalized nonlinear models) is the poisson regression. According to G.James (James et al.) this kind of nonlinear regression is basically similar to the well-known regular multiple regression, except this time the dependent variable (Y) has observations that follow the poisson (and not the normal) distribution and their numbers are strictly positive (and this is the a characteristic that distinguish normal from poisson distribution).

To be more specific, according to Dataquest ("Learn to Use Poisson Regression in R", 2019), poisson distribution is a very unique category of data distribution and it can be very useful if our task is to measure the times, or the possibilities, that an incident occurs during a specifically given timeline. Moreover, in case that the dataset we use has categorical independent variables, poisson model tends to reform them into a set of 0 and 1, due to the fact that poisson models are able to operate only if the variable set contains numbers, or in other words the dataset is defined by count or discrete data, and in case that any independent variable has missing values, its observation row will be elided from the dataset.

Last but not least, in comparison with the traditional logistic model, poisson regression does not result only in binary output, but it is also able to make legit predictions for a discrete dependent variable. This extraordinary model which combines characteristics from the linear and the logistic regression models, seems to be a very useful tool in cases which the prediction of a rare outcome is very important for us and for this reason it's being used widely in the healthcare sector.

Polynomial regression

Red wines

- We Build The Simple Linear Regression And Then We Build Four More Polynomial Regressions

  Comment:
  In order to conduct a polynomial analysis we had to create a simple linear regression between quality and alcohol (alcohol is the independent variable that has the highest correlation (0.476) with quality) and then we had to create four different polynomial regressions in which the independent variable had exponents (the exponents range from second degree to fourth degree depending the regression).

- We Find Out Which Is The Most Appropriate Degree For Our Polynomial Model

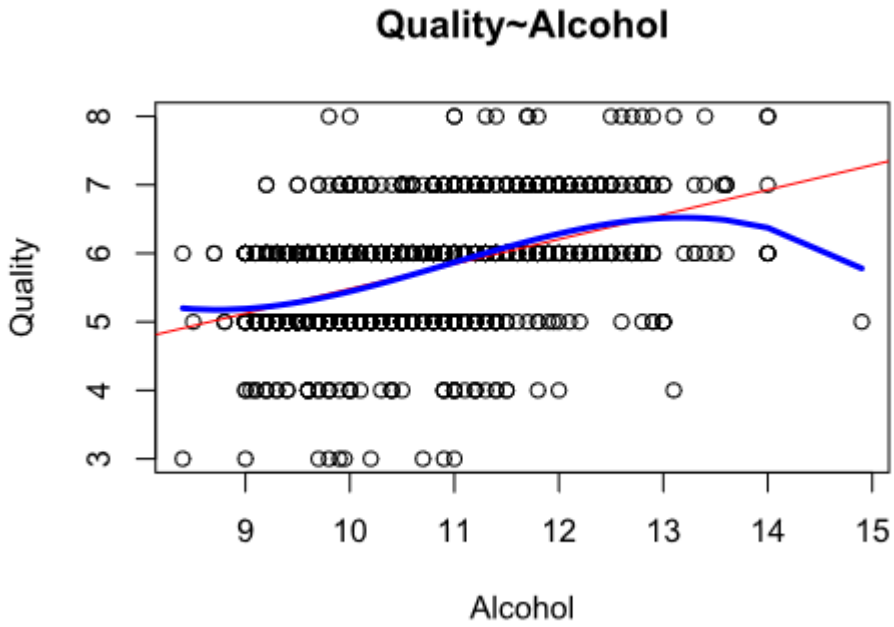
  Comment:
  Through an Anova analysis that has been conducted between our simple regression and the remaining four polynomial models, the results showed that the third model (fit3), which is the third degree polynomial regression, is the most suitable for our dataset. This conclusion has been made after a comparison between our models' p.values. This comparison was able to show that fit3 was the only model that had a p.value smaller than 0.05 (which is the significance frontier).Also, our main analysis shows, that the intercept is b0=40.638. Furthermore b1=-10.428, b3=0.993, b3=-0.030and there is an R-squared (multiple and adjusted) around 23% (0.2341 and 0.2327) that shows low data concentration levels around our fitted regression line.
  Quality= 40.638 - 10.428*Alcohol + 0.992*Alcohol^2 -0.030*Alcohol^3.

- We Separate The Whole Dataset In Training And Testing Set (70-30%)

- Prediction Metrics

PREDICION METRICS	VALUES
RMSE	0.865
MAE	0.700
MAPE	0.129
MSE	0.748
MIN MAX ACCURACY	0.885

- We Create The Appropriate Plot



Comment:
From the plot above, we notice that the polynomial regression analysis may be a problematic one for a dataset like ours, in which the dependent variable has discrete observations. The red line represents the simple linear regression (fit) and the blue line represents our final polynomial model (fit3). As we can see, this special formation of our observations can cause a data-fit problem, due to the fact that none of these two regression lines is a good fit for our data. For this reason this specific type of regression analysis could be inappropriate for our analysis and it is possible that its prediction metrics could be misleading.

Logistic regression

Red wines

- We Create A Factor Variable And We Update Our Dataset

  Comment:
  In order to run a logistic regression it is important to have a factor variable. This new variable has to be our new dependent variable and in our case we created "factor.quality" which categorizes wine qualities in two levels (good and bad) depending their score and replaced it with our previous dependent variable (Quality). In this new factor variable, every wine that has an overall score below 5 is considered as a bad wine and every wine that has an overall score above 5 is considered as a good wine.

- We Build The Logistic Regression Model

  Comment:
  The main analysis shows, that the intercept is b0=42.950. Furthermore b1= 0.136, b2=-3.282, b3=-1.274, b4= 0.055, b5=-3.916, b6=0.022, b7= -0.016, b8= -50.932, b9= -0.381, b10= 2.795 and b11=0.867.
  factor.quality= 42.950 + 0.136*FixedAcidity – 3.282*VolatileAcidity - 1.274*CitricAcid + 0.055*ResidualSugar - 3.915*Chlorides + 0.022*FreeSulfurDioxide - 0.016*TotalSulfurDioxide -50.932*Density - 0.380*PH + 2.795*Sulphates + 0.867*Alcohol.

- We Predict The Probabilities' Accuracy For Having Good And Bad Wines (Using The Whole Dataset)

GLM.PROBABILITIES	BAD QUALITY	GOOD QUALITY
0	549	214
1	195	641

PREDICTION METRIC	VALUES
ACCURACY	0.7442151

Comment:
From the analysis we notice that our model's correct predictions (the diagonal elements) have a percentage of (549+641)/1599=0.744%. This result indicates that the logistic regression correctly predicted the movement of our wines' quality 74.4% of the time (641 correct predictions for good quality wines and 549 correct predictions for bad quality wines). Although this prediction may be misleading, because it

was conducted using the whole dataset. For this reason we have to separate our dataset in training and testing set and run again our prediction functions.

- We Separate The Whole Dataset In Training And Testing Set (70-30%)

- We Predict The Probabilities' Accuracy For Having Good And Bad Wines (Using The Training And Testing Set)

GLM.PROBABILITIES	BAD QUALITY	GOOD QUALITY
0	180	71
1	54	175

PREDICTION METRIC	VALUES
ACCURACY	**0.739**

Comment:
From this table we notice that our model's correct predictions (the diagonal elements) have a percentage of (180+175)/480=0.739%. This result indicates that the logistic regression correctly predicted the movement of our wines' quality 73.9% of the time (175 correct predictions for good quality wines and 180 correct predictions for bad quality wines). It appears that this percentage is very close to our previous one, but what should it be if we throw away our insignificant variables?

- We Build Again The Logistic Regression Model (Without The Insignificant Variables)

Comment:
From the analysis we can notice that the intercept is b0=-8.150. Furthermore b1= -2.896, b2=-4.421, b3=0.024, b4= -0.018, b5=2.706, b6=0.859.
factor.quality= -8.150 – 2.896*VolatileAcidity – 4.421*Chlorides + 0.024*FreeSulfurDioxide – 0.018*TotalSulfurDioxide + 2.706*Sulphates + 0.859*Alcohol.

- We Separate The New Dataset In Training And Testing Set (70-30%)

- We Predict The Probabilities' Accuracy For Having Good And Bad Wines (Using The Training And Testing Set) And We Gain The Prediction Metric

GLM.PROBABILITIES	BAD QUALITY	GOOD QUALITY
0	184	70
1	50	176

PREDICTION METRIC	VALUES
ACCURACY	0.75

Comment:
From this table we notice that our model's correct predictions (the diagonal elements) have a percentage of (184+176)/480=0.75%. This result indicates that the logistic regression correctly predicted the movement of our wines' quality 75% of the time (176 correct predictions for good quality wines and 184 correct predictions for bad quality wines). It appears that this percentage is very close to our previous results, but at the same time it's the best one. This prediction accuracy shows that in logistic regression it is important to keep the most significant variables within the model in order to have the best prediction possibilities.

Poisson regression

Red wines

- We Build The Poisson Regression Model

Comment:
From the analysis we can notice that the intercept is b0=38.621.Furthermore b1= 1.004, b2=0.821, b3=0.965, b4=1.003, b5=0.718, b6=1.001, b7=0.999, b8=0.114, b9=0.928, b10=1.172 and b11=1.049.
Quality= 38.621 + 1.004*FixedAcidity + 0.821*VolatileAcidity +0.965*CitricAcid +1.003*ResidualSugar +0.718*Chlorides + 1.001*FreeSulfurDioxide + 0.999*TotalSulfurDioxide + 0.114*Density + 0.928*PH +1.172*Sulphates + 1.049*Alcohol.
(note: the numbers have been adjusted in order to interpret the regression coefficients in the original scale of the dependent variable (Quality) and not in the log one).

- We Build Again The Poisson Regression Model (Without The Insignificant Variables)

Comment:
The main analysis shows, that the intercept is b0=38.621.Furthermore b1= 1.004 and b2=0.821.
Quality= 3.658 + 0.780*VolatileAcidity + 1.055*Alcohol.
(note: the numbers have been adjusted in order to interpret the regression coefficients in the original scale of the dependent variable (Quality) and not in the log one).

- We Check For Overdispersion

VARIABLE	**QCC.OVERDISPERSION TEST VALUES (p.value)**
QUALITY	1

Comment:
From the table above we notice that there is no overdispersion (p.value is bigger than 0.05) in our dependent variable (Quality) which is good for our model.
(note: it is very important to check for overdispersion, because in Poisson regression, the variance and means are equal. Overdispersion occurs when the observed variance of the response variable is larger than would be predicted by the Poisson distribution).

- We Separate The Whole Dataset In Training And Testing Set (70%-30%)
- Prediction Metrics

PREDICION METRICS	VALUES
RMSE	3.922
MAE	3.849
MAPE	0.684
MSE	15.387
MIN MAX ACCURACY	0.315

Chapter 5

Decision tree and random forest regressions

Theoretical background

According to G. James (James et al.),tree-based prediction methods can be applied both in regression and classification problems. They are called tree methods, due to the fact that in order to predict the result of a specific variable, they split the whole variable space in several tree look-alike categories and compare each result using a training and a testing dataset.

This type of prediction methods is considered being very simple and quite useful in many cases, but also they tend to lack in prediction accuracy compared to our previous regression models. For this reason decision tree regression model, which was the first one of its kind, gave birth to random forest regression that uses a large number of various decision tree in order to produce a better and more accurate prediction result than its predecessor.

For us to be more precise, decision tree works by building a top-down tree using a method that's called recursive binary splitting. Through this method it splits the predictor space starting from the top of the tree and in each stock it produces a pair of two new branches. This splitting method is known as a greedy one, because at each step tries to create the best branch at the time being, rather than foreseeing and choosing a different split that will lead to a better and more accurate, considering its prediction abilities, tree.

In order to overcome decision tree's accuracy problem, scientists developed the tree pruning method. Tree pruning is basically an elegant way to end up with a tree that has low variance level in expense of some lower bias. It works by creating a very large tree and then we prune it until we have our final subtree. However, this method isn't the best choice if we want to obtain the most reliable and accurate prediction results, due to the fact that even if it produces good results in our training set, it leads to poor ones in the testing set, because most of the time the resulting subtree might be too complicated.

For this reason, many different tree-based methods have been created along with the random forest one. Random forest regression provides a better prediction accuracy by building a team of several decision tree on bootstrapped training dataset samples. During this process, each split leads to a random sample collection of m predictors which works basically as split candidates from the whole set of p predictors. This split uses only just one of these m predictors and a brand new sample of their kind is taken at each split. This type of regression process may sounds complicated and crazy, but it is very effective due to the fact that in cases that there is a very strong predictor in our dataset, then the bagged decision tree will use this one as the top predictor in every split, leading to a total amount of tree that would look the same and so there would be a high correlation in our prediction results. Nevertheless this is not the main reason that random forest overcome the flaws of the single decision tree regression. In order to succeed a lower

variance that would improve the prediction accuracy, random forest forces each split of the regression to consider only a subset of the predictors, giving the chances to other (more moderately powerful) ones to take the top position in our single decision tree that are part of the forest. This incredibly smart type of tree-based regression has become very popular nowadays and the main reason is that is able to cope with the lack of prediction accuracy that we may face in a single decision tree regression.

Decision tree regression

Red wines

- We Create A Factor Variable And We Update Our Dataset

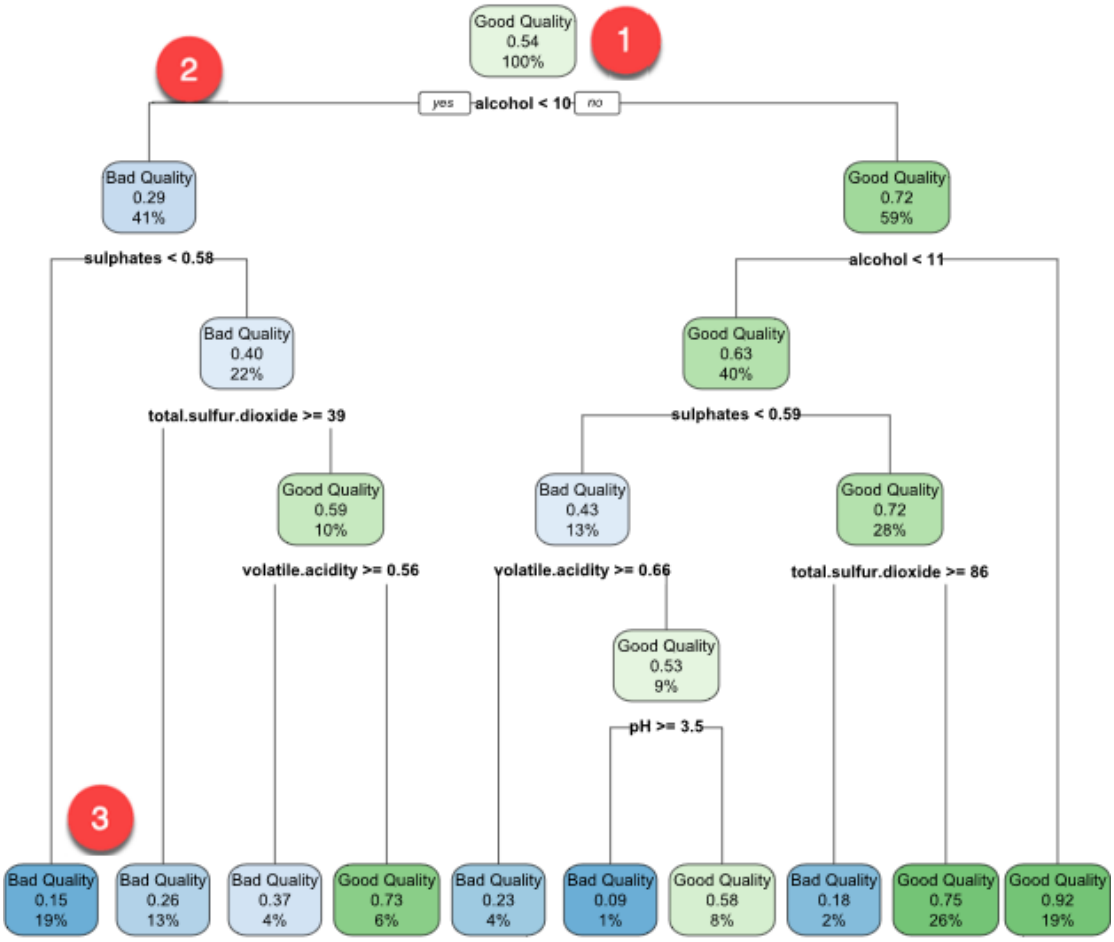
  Comment:
  In order to run a decision tree regression it is important to have a factor variable. This new variable has to be our new dependent variable and in our case we created "factor.quality" which categorizes wine qualities in two levels (good and bad) depending their score and replaced it with our previous dependent variable (Quality). In this new factor variable, every wine that has an overall score below 5 is considered as a bad wine and every wine that has an overall score above 5 is considered as a good wine.

- We Shuffle The Dataset

  Comment: In order to have a representative sample of our whole dataset in our training and testing sets we shuffle the observations (this isn't necessary in our case, because the observations are already shuffled but it is an opportunity to try this specific code function).

- We Separate The Whole Dataset In Training And Testing Set (70%-30%)

- We Build The Decision Tree Model And We Gain A Visual Representation



Comment:

1. At the top, it is the overall probability of having a good wine as a result. It shows the proportion of wines that turned to have a good quality (overall score above 5). 54 percent of the wines are good.
2. This node asks whether the alcohol concentration is below 10%. If yes, then you go down to the root's left child node (depth 2). 41 percent have alcohol concentration below 10% a good wine probability of 29 percent.
3. In the second node, you ask if the sulphates concentration is below 0.58. If yes, then the chance of having a good quality wine is 15 percent.
4. You keep on going like that to understand what features impact the likelihood of having a good wine.

   As we notice from the diagram above, the most important element in order to result with a good quality wine is alcohol. 92% of the times when the alcohol concentration was above 11%, have resulted in quality

overall scores bigger than 5 out of 10 (in other words the bigger the alcohol concentration the better the wine quality).

- We Predict The Probabilities' Accuracy For Having Good And Bad Wines (Using The Training And Testing Set) And We Gain The Prediction Metric

GLM.PROBABILITIES	BAD QUALITY	GOOD QUALITY
0	161	73
1	62	184

PREDICTION METRIC	VALUES
ACCURACY	0.718

Comment:
From these tables we notice that our model's correct predictions (the diagonal elements) have a percentage of (161+184)/480=0.718%. This result indicates that the decision tree regression correctly predicted the movement of our wines' quality 71.8 % of the time (184 correct predictions for good quality wines and 161 correct predictions for bad quality wines).

Random forest regression

Red wines

- We Create A Factor Variable And We Update Our Dataset

  Comment:
  In order to run a decision tree regression it is important to have a factor variable. This new variable has to be our new dependent variable and in our case we created "factor.quality" which categorizes wine qualities in two levels (good and bad) depending their score and replaced it with our previous dependent variable (Quality). In this new factor variable, every wine that has an overall score below 5 is considered as a bad wine and every wine that has an overall score above 5 is considered as a good wine.

- We Separate The Whole Dataset In Training And Testing Set (70%-30%)

- We Train The Regression Using The Grid Search Method In Order To Find The Best Parameter Combination For Our Model
  Comment:
  In order to end up with the most sufficient parameters that will generalize best our data and will give us the best prediction results, we run the grid search that will help us find the best "mtry" (number of candidates draw to feed the algorithm. By default, it is the square of the number of columns), "ntree" (number of tree in the forest) and "maxnodes" (set the maximum amount of terminal nodes in the forest) that are essential parameters for this specific regression type.

- We End Up With The Best Parameters

MTRY	**4**
NTREE	**350**
MAXNODES	**5**

Comment:
The grid search resulted in the most suitable combination of our parameters that will lead us in the most accurate and reliable prediction results. In order to find out which ones were the best parameters, accuracy was the key. Having conducted a grid search the accuracy of the model for each one of our parameters (mtry, maxnodes, ntree) was 0.7988739, 0.8125000 and 0.7946429 respectively.

- We Build The Model Using The Best Parameters

- We Predict The Probabilities' Accuracy For Having Good And Bad Wines (Using The Training And Testing Set) And We Gain The Prediction Metric

GLM.PROBABILITIES	BAD QUALITY	GOOD QUALITY
0	159	66
1	75	180

PREDICTION METRIC	VALUES
ACCURACY	**0.706**

Comment:
From these tables we notice that our model's correct predictions (the diagonal elements) have a percentage of (159+180)/480=0.706%. This result indicates that the decision tree regression correctly predicted the movement of our wines' quality 70.6 % of the time (180 correct predictions for good quality wines and 159 correct predictions for bad quality wines).

Conclusion

After having conducted eight different regression models over a discrete dataset that consists of 12 variables and 1599 observations, it's time for us to come to a conclusion.

Since the beginning of our analysis we noticed that our red wine dataset in which "Quality" was the dependent variable had both negative and positive types of correlation between its variables. To be more specific, our dependent variable is negatively correlated to six independent variables (volatile.acidity, chlorides, free.sulfur.dioxide, total.sulfur.dioxide, density and Ph) and positively correlated to the remaining five (fixed.acidity, citric.acid, residual.sugar, sulphates and alcohol) with volatile.acidity and alcohol sharing the two strongest correlations (negative and positive respectively) with Quality. These relations were decisive for the development of our project as we are going to explain in the next paragraphs.

The first regression method in which we applied our precious dataset, was the multiple linear regression. This famous statistical model didn't let us down. After having fulfilled almost every one of its assumptions (except the existence of homoscedasticity) it ended up with six statistical significant independent variables and one of the highest prediction accuracy percentage (88.3%) in our whole analysis. The only drawback that can be spotted in this prediction type is the fact that the remaining prediction metrics (RMSE, MSE, MAPE, MAE) had also high values, which means that there is a possibility for its predictions to be misleading and unreliable. Moving into the same direction, the following two prediction methods, that were ridge and lasso regressions, had positive results as well. These two pioneer methods, even though they work without separating the variables in significant and insignificant ones, showed to work as good as our previous linear type, with lasso having a little bit better results in its prediction metrics. However, just like ML both of them had quite high values in their prediction errors, meaning that they could end up in false results considering their prediction accuracy.

The next three regressions that we examined belong in the famous generalized non-linear family. Starting with the first one, polynomial regression used alcohol as its only independent variable and after an Anova analysis it turned out that the most suitable form for this model was the one in which alcohol was raised in the 3^{rd} power. It's prediction accuracy was very high (88.5%) but the bad remaining metrics in addition with the fact that this type of prediction method works better on continuous and not on discrete datasets, tend to make this regression unstable for wine quality predictions. Moving on the second type of this family, logistic regression showed to be a lot promising. In contrast to every previous method, it needs a factor variable in the position of the dependent variable in order to work. For this reason, factor.quality was created for the first time and it is a variable that categorizes wine quality in good and bad according to their overall quality score (good quality>5, bad quality<5). Logistic regression selected six

significant independent variables and had a high prediction accuracy of 75% without showing anything problematic. However, the last model of this category didn't work as good as its predecessor. Poisson regression which is very similar to our previous logistic model had the worst prediction results in our analysis having only two independent variables in its regression, making crystal clear that the absence of a factor variable in this kind of regression types has catastrophic results in the prediction sector.

Last but not least, we examined two tree-based methods that are very famous in the statistics world for their ability in making predictions very easily. Both decision tree regression and its advanced type of random forest regression work by having a factor variable (factor.quality) in the position of the dependent variable, just like logistic regression does. Once again, these models showed that alcohol plays a significant role in our analysis, due to the fact that it is the component which is responsible for having the biggest impact on wine quality. These tree-based methods were very useful for a dataset like ours and they both had high prediction accuracy percentages (71.8% and 70.6% respectively) making them two of the most suitable candidates for a wine quality regression analysis.

To sum up, every single regression type that we examined using our discrete dataset had something to tell us. Traditional methods like the linear regressions did well but almost all of them had high error possibilities that affect their prediction accuracy making them unreliable and unstable for safe predictions. Nevertheless, this paper showed that three of our regression models (logistic, decision tree and random forest) did more than well in predicting the quality of our wines, choosing alcohol as the most critical component and making us believe that using regression methods that work only with a dependent factor variable is the right thing to do in discrete datasets like this one.

Summary of prediction metrics

Multiple linear regression

PREDICION METRICS	VALUES
RMSE	0.902
MAE	0.721
MAPE	0.132
MSE	0.813
MIN MAX ACCURACY	0.883

10 folders cross validation with MLR

METRICS	RMSE	MAE
70-30 %	0.902	0.721
10 folders cross validation	0.651	0.504

Ridge regression

PREDICTION METRICS	VALUES
MSE	0.638
MAE	0.499
RMSE	0.638
MAPE	0.091

Lasso regression

PREDICTION METRICS	VALUES
MSE	0.412
MAE	0.505
RMSE	0.642
MAPE	0.092

Polynomial regression

PREDICION METRICS	VALUES
RMSE	0.865
MAE	0.700
MAPE	0.129
MSE	0.748
MIN MAX ACCURACY	0.885

Logistic regression

PREDICTION METRIC	VALUES
ACCURACY	0.75

Poisson regression

PREDICION METRICS	VALUES
RMSE	3.922
MAE	3.849
MAPE	0.684
MSE	15.387
MIN MAX ACCURACY	0.315

Decision tree regression

PREDICTION METRIC	VALUES
ACCURACY	0.718

Random forest regression

PREDICTION METRIC	VALUES
ACCURACY	0.706

Appendix

Red Wines Dataset Script (Dataset Analysis And Multiple Linear Regression)

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("ggplot2")
library(ggplot2)
install.packages("psych")
library(psych)
install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
install.packages("corrplot")
library(corrplot)
install.packages("FactoMineR")
library(FactoMineR)
install.packages("lmtest")
library(lmtest)
install.packages("normtest")
library(normtest)
install.packages("nortest")
library(nortest)
install.packages("car")
library(car)
install.packages("dgof")
library(dgof)
install.packages("bestNormalize")
library(bestNormalize)
install.packages("leaps")
library(leaps)
install.packages("car")
library(car)
install.packages("lmtest")
library(lmtest)
install.packages("MLmetrics")
library(MLmetrics)
install.packages("caret")
library(caret)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE MAKE THE GRAPHICAL ANALYSIS OF THE DATASETS
# WITH HISTOGRAMS
qplot(mydatasetred1$quality, geom="histogram", binwidth = 1,main =
"Quality",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$fixed.acidity, geom="histogram", binwidth =
1,main = "Fixed Acidity",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$volatile.acidity, geom="histogram", binwidth =
0.1,main = "Volatile Acidity",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
```

```
qplot(mydatasetred1$citric.acid, geom="histogram", binwidth =
0.1,main = "Citric Acid",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$residual.sugar, geom="histogram", binwidth =
1,main = "Residual Sugar",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$chlorides, geom="histogram", binwidth = 0.03,main
= "Chlorides",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$free.sulfur.dioxide, geom="histogram", binwidth =
3,main = "Free Sulfur Dioxide",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$total.sulfur.dioxide, geom="histogram", binwidth
= 10,main = "Total Sulfur Dioxide",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$density, geom="histogram", binwidth = 0.001,main
= "Density",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$pH, geom="histogram", binwidth = 0.1,main = "PH",
xlab = "scores",ylab = "observations",fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$sulphates, geom="histogram", binwidth = 0.1,main
= "Sulphates",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
qplot(mydatasetred1$alcohol, geom="histogram", binwidth = 0.5,main =
"Alcohol",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
# WITH BOXPLOTS
par(mfrow = c(1,2))
boxplot(mydatasetred1$fixed.acidity, main = "Fixed Acidity",
horizontal = T,  xlab= "observations", col = c("red"))
summary(mydatasetred1$fixed.acidity)
boxplot(mydatasetred1$volatile.acidity, main = "Volatile
Acidity",horizontal = T,  xlab= "observations", col = c("red") )
summary(mydatasetred1$volatile.acidity)
boxplot(mydatasetred1$citric.acid, main = "Citric Acid", horizontal =
T,  xlab= "observations", col = c("red"))
summary(mydatasetred1$citric.acid)
boxplot(mydatasetred1$residual.sugar, main = "Residual Sugar",
horizontal = T,  xlab= "observations", col = c("red"))
summary(mydatasetred1$residual.sugar)
boxplot(mydatasetred1$chlorides, main = "Chlorides", horizontal = T,
xlab= "observations", col = c("red"))
summary(mydatasetred1$chlorides)
boxplot(mydatasetred1$free.sulfur.dioxide, main = "Free Sulfur
Dioxide", horizontal = T,  xlab= "observations", col = c("red"))
summary(mydatasetred1$free.sulfur.dioxide)
boxplot(mydatasetred1$total.sulfur.dioxide, main = "Total Sulfur
Dioxide", horizontal = T,  xlab= "observations", col = c("red"))
summary(mydatasetred1$total.sulfur.dioxide)
boxplot(mydatasetred1$density, main = "Density", horizontal = T,
xlab= "observations", col = c("red"))
```

```
summary(mydatasetred1$density)
boxplot(mydatasetred1$pH, main = "PH", horizontal = T,  xlab=
"observations", col = c("red"))
summary(mydatasetred1$pH)
boxplot(mydatasetred1$sulphates, main = "Sulphates", horizontal = T,
xlab= "observations", col = c("red"))
summary(mydatasetred1$sulphates)
boxplot(mydatasetred1$alcohol, main = "Alcohol", horizontal = T,
xlab= "observations", col = c("red"))
summary(mydatasetred1$alcohol)
boxplot(mydatasetred1$quality, main = "Quality", horizontal = T,
xlab= "observations", col = c("red"))
summary(mydatasetred1$quality)
# WE FIND THE OUTLIERS OF EACH VARIABLE
boxplot.stats(mydatasetred1$fixed.acidity) # 50 outliers (3%)
boxplot.stats(mydatasetred1$volatile.acidity) # 20 outliers (1.2%)
boxplot.stats(mydatasetred1$citric.acid) # 1 outlier (0%)
boxplot.stats(mydatasetred1$residual.sugar) #156 outliers (9%)
boxplot.stats(mydatasetred1$chlorides) #113 outliers (7%)
boxplot.stats(mydatasetred1$free.sulfur.dioxide) #30 outliers (1.8%)
boxplot.stats(mydatasetred1$total.sulfur.dioxide) #56 outliers (3.6%)
boxplot.stats(mydatasetred1$density) #46 outliers (2.8%)
boxplot.stats(mydatasetred1$pH) #36 outliers (2.2%)
boxplot.stats(mydatasetred1$sulphates) #60 outliers (3.7%)
boxplot.stats(mydatasetred1$alcohol) #14 outliers (0.8%)
boxplot.stats(mydatasetred1$quality) #28 outliers (1.7%)
# WE CHECK FOR CORRELATION
par(mfrow = c(1,1))
mycor<- round(cor(mydatasetred1), 3)
mycor
mydatasetred1a <- mydatasetred1[ ,c(1,2,3,4,5,6,12)]
mydatasetred1b <- mydatasetred1[ ,c(7,8,9,10,11,12)]
pairs(mydatasetred1a) #1st WAY
pairs(mydatasetred1b) #1st WAY
pairs.panels(mydatasetred1a)#2nd WAY
pairs.panels(mydatasetred1b)#2nd WAY
chart.Correlation(mydatasetred1a, histogram = T, pch = 19) #3rd WAY
chart.Correlation(mydatasetred1b, histogram = T, pch = 19) #3rd WAY
mycor<- cor(mydatasetred1) #4th WAY
corrplot(mycor, type = "upper", method = "pie")
plot_matrix <- function(matrix_toplot){
corrplot.mixed(matrix_toplot,
order = "original",
                  tl.col='black', tl.cex=.50)
}
plot_matrix(cor(mydatasetred1)) #5th WAY
# PCA
res.pca <- PCA(mydatasetred1, graph = T); res.pca
# WE BUILD A SIMPLE REGRESSION BETWEEN QUALITY AND ALCOHOL
simpleregred<- lm(mydatasetred1$quality ~ mydatasetred1$alcohol)
summary(simpleregred)
plot(mydatasetred1$quality ~ mydatasetred1$alcohol, main = "Quality ~
Alcohol", xlab = "Alcohol", ylab = "Quality", col = c("red"))
# WE CHECK FOR THE SIMPLE REGRESSION ASSUMPTIONS
# No1 WE CHECK FOR LINEARITY
plot(simpleregred$residuals, xlab = "Alcohol", main = "Simple
Regression Residuals")
# No2 WE CHECK FOR HOMOSCEDASTICITY
bptest(simpleregred)#P.VALUE < 0.05 SO WE HAVE HETEROSCEDASTICITY
# No3 WE CHECK FOR CORRELATION BETWEEN X VARIABLE AND THE RESIDUALS
```

```
cor.test(mydatasetred1$alcohol, simpleregred$residuals)
# No4 WE CHECK FOR NORMALITY IN THE RESIDUALS
qplot(simpleregred$residuals, geom="histogram", binwidth = 0.2,main =
"Residuals",
xlab = "scores",fill=I("blue"), col=I("red"),alpha=I(.2),)
boxplot(simpleregred$residuals, main = "residuals", horizontal = T,
xlab= "observations", col = c("red") )
boxplot.stats(simpleregred$residuals)
qqnorm(simpleregred$residuals)
qqline(simpleregred$residuals)
shapiro.test(simpleregred$residuals) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
jb.norm.test(simpleregred$residuals) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
ad.test(simpleregred$residuals)  #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
cvm.test(simpleregred$residuals) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
ks.test(simpleregred$residuals,"pnorm", mean(simpleregred$residuals),
sd(simpleregred$residuals) ) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
res.student <- rstudent(simpleregred)
res.student
plot(res.student,pch= 15, cex= .5,ylab = "Studentized Residuals")
abline(h=c(-2,0,2), lty= c(2,1,2))
res.standard <- rstandard(simpleregred)
res.standard
plot(res.standard, phc = 15, cex = .5,ylab = "Standarized Residuals")
abline(h=c(-2,0,2), lty= c(2,1,2))
# (5) WE CHECK FOR AUTOCORRELATION
durbinWatsonTest(simpleregred) #P.VALUE < 0.05 SO WE HAVE
AUTOCORRELATION
# WE CHECK FOR NORMALITY IN THE DEPENDENT VARIABLE
shapiro.test(mydatasetred1$quality)#p.value < 0.05 SO THERE IS NO
NORMALITY IN THE DEPENDENT VARIABLE
jb.norm.test(mydatasetred1$quality) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
ad.test(mydatasetred1$quality)  #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE DEPENDENT VARIABLE
cvm.test(mydatasetred1$quality) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE DEPENDENT VARIABLE
ks.test(mydatasetred1$quality,"pnorm", mean(mydatasetred1$quality),
sd(mydatasetred1$quality) ) #p.value < 0.05 SO WE DONT HAVE NORMALITY
IN THE DEPENDENT VARIABLE
# WE TRANSFORM THE DEPENDENT VARIABLE AND THEN WE CHECK AGAIN FOR
NORMALITY
BNQuality <- bestNormalize(mydatasetred1$quality) #1st way
BNQuality
logquality<- log(mydatasetred1$quality) #2nd way
qplot(logquality, geom="histogram", binwidth = 0.16,main =
"LogQuality",
xlab = "scores", ylab = "observations", fill=I("blue"),
col=I("red"),alpha=I(.2),)
shapiro.test(logquality)#p.value < 0.05 SO THERE IS NO NORMALITY IN
THE DEPENDENT VARIABLE
jb.norm.test(logquality) #p.value < 0.05 SO WE DONT HAVE NORMALITY IN
THE DEPENDENT VARIABLE
ad.test(logquality)  #p.value < 0.05 SO WE DONT HAVE NORMALITY IN THE
DEPENDENT VARIABLE
```

```
cvm.test(logquality) #p.value < 0.05 SO WE DONT HAVE NORMALITY IN THE
DEPENDENT VARIABLE
ks.test(logquality,"pnorm", mean(logquality), sd(logquality) )
#p.value < 0.05 SO WE DONT HAVE NORMALITY IN THE DEPENDENT VARIABLE
# WE GET VARIANCE, STANDARD DEVIATION AND STANDARD ERROR
# R-->RedWine
RVarFixedAcidity <- var(mydatasetred1$fixed.acidity)
RVarFixedAcidity
RVarVolatileAcidity <- var(mydatasetred1$volatile.acidity)
RVarVolatileAcidity
RVarCitricAcid <- var(mydatasetred1$citric.acid)
RVarCitricAcid
RVarResidualSugar <- var(mydatasetred1$residual.sugar)
RVarResidualSugar
RVarChlorides <- var(mydatasetred1$chlorides)
RVarChlorides
RVarFreeSulfurDioxides <- var(mydatasetred1$free.sulfur.dioxide)
RVarFreeSulfurDioxides
RVarTotalSulfurDioxide <- var(mydatasetred1$total.sulfur.dioxide)
RVarTotalSulfurDioxide
RVarDensity <- var(mydatasetred1$density)
RVarDensity
RVarPH <- var(mydatasetred1$pH)
RVarPH
RVarSulphates <- var(mydatasetred1$sulphates)
RVarSulphates
RVarAlcohol <- var(mydatasetred1$alcohol)
RVarAlcohol
RVarQuality <- var(mydatasetred1$quality)
RVarQuality
RSDFixedAcidity <- sd(mydatasetred1$fixed.acidity)
RSDFixedAcidity
RSDVolatileAcidity <- sd(mydatasetred1$volatile.acidity)
RSDVolatileAcidity
RSDCitriAcid <- sd(mydatasetred1$citric.acid)
RSDCitriAcid
RSDResidualSugar <- sd(mydatasetred1$residual.sugar)
RSDResidualSugar
RSDChlorides <- sd(mydatasetred1$chlorides)
RSDChlorides
RSDFreeSulfurDioxide <- sd(mydatasetred1$free.sulfur.dioxide)
RSDFreeSulfurDioxide
RSDTotalSulfurDioxide <- sd(mydatasetred1$total.sulfur.dioxide)
RSDTotalSulfurDioxide
RSDDensity <- sd(mydatasetred1$density)
RSDDensity
RSDPH <- sd(mydatasetred1$pH)
RSDPH
RSDSulphates <- sd(mydatasetred1$sulphates)
RSDSulphates
RSDAlcohol <- sd(mydatasetred1$alcohol)
RSDAlcohol
RSDQuality <- sd(mydatasetred1$quality)
RSDQuality
RSEFixedAcidity <-
RSDFixedAcidity/sqrt(length(mydatasetred1$fixed.acidity))
RSEFixedAcidity
RSEVolatileAcidity <-
RSDVolatileAcidity/sqrt(length(mydatasetred1$volatile.acidity))
RSEVolatileAcidity
```

```
RSECitricAcid <- RSDCitriAcid/sqrt(length(mydatasetred1$citric.acid))
RSECitricAcid
RSEResidualSugar <-
RSDResidualSugar/sqrt(length(mydatasetred1$residual.sugar))
RSEResidualSugar
RSEChlorides <- RSDChlorides/sqrt(length(mydatasetred1$chlorides))
RSEChlorides
RSEFreeSulfurDioxide <-
RSDFreeSulfurDioxide/sqrt(length(mydatasetred1$free.sulfur.dioxide))
RSEFreeSulfurDioxide
RSETotalSulfurDioxide <-
RSDTotalSulfurDioxide/sqrt(length(mydatasetred1$total.sulfur.dioxide)
)
RSETotalSulfurDioxide
RSEDensity <- RSDDensity/sqrt(length(mydatasetred1$density))
RSEDensity
RSEPH <- RSDPH/sqrt(length(mydatasetred1$pH))
RSEPH
RSESulphates <- RSDSulphates/sqrt(length(mydatasetred1$sulphates))
RSESulphates
RSEAlcohol <- RSDAlcohol/sqrt(length(mydatasetred1$alcohol))
RSEAlcohol
RSEQuality <- RSDQuality/sqrt(length(mydatasetred1$quality))
RSEQuality
# WE BUILT THE MULTIPLE LINEAR REGRESSION WITH EVERY SINGLE
INDEPENDENT VARIABLE
multilinearregred <- lm(mydatasetred1$quality ~
mydatasetred1$fixed.acidity+mydatasetred1$volatile.acidity+mydatasetr
ed1$citric.acid+mydatasetred1$residual.sugar+mydatasetred1$chlorides+
mydatasetred1$free.sulfur.dioxide+mydatasetred1$total.sulfur.dioxide+
mydatasetred1$density+mydatasetred1$pH+mydatasetred1$sulphates+mydata
setred1$alcohol)
summary(multilinearregred)
# DECIDING ON IMPORTANT VARIABLES WITH FORWARD, BACKWARD AND STEPWISE
METHODS
# REGSUBSETS FORMULA
planA<- regsubsets(quality~., data = mydatasetred1, nbest = 1, nvmax
= 12, method = "forward")
plot(planA, scale= "bic")
summary(planA)
coef(planA, which.min(summary(planA)$bic))
planB<- regsubsets(quality~., data = mydatasetred1, nbest = 1, nvmax
= 12, method = "backward")
plot(planB, scale= "bic")
summary(planB)
coef(planB, which.min(summary(planB)$bic))
planC<- regsubsets(quality~., data = mydatasetred1, nbest = 1, nvmax
= 12)
plot(planC, scale= "bic")
summary(planC)
coef(planC, which.min(summary(planC)$bic))
# WE KEEP 6 VARIABLES (VOLATILE ACIDITY, CHLORIDES, TOTAL SULFUR
DIOXIDE, PH, SULPHATES, ALCOHOL)
# STEP FORMULA
FitAll <- lm(quality~., data=mydatasetred1)
FitStart <- lm(quality~1, data=mydatasetred1)
step(FitStart, direction="forward", scope=formula(FitAll)) #forward
step(FitAll, direction= "backward", trace=F) #backward
step(FitStart, direction="both", scope = formula(FitAll)) #stepwise
# FINAL REGRESSIONS
```

```
finmultilinearregred<- lm(mydatasetred1$quality ~
mydatasetred1$volatile.acidity+mydatasetred1$chlorides+mydatasetred1$
total.sulfur.dioxide+mydatasetred1$pH+mydatasetred1$sulphates+mydatas
etred1$alcohol)
summary(finmultilinearregred)
# WE CHECK FOR THE MULTIPLE REGRESSION ASSUMPTIONS
# No1 WE CHECK FOR MULTICOLLINEARITY IN THE FINAL REGRESSION
vif(finmultilinearregred)
# No2 WE CHECK FOR NORMALITY IN THE RESIDUALS OF THE FINAL REGRESSION
par(mfrow =c(1,1))
qplot(finmultilinearregred$residuals, geom="histogram", binwidth =
0.2,main = "Residuals",
xlab = "scores",fill=I("blue"), col=I("red"),alpha=I(.2),)
boxplot(finmultilinearregred$residuals, main = "residuals",
horizontal = T,  xlab= "observations", col = c("red") )
boxplot.stats(finmultilinearregred$residuals)
qqnorm(finmultilinearregred$residuals)
qqline(finmultilinearregred$residuals)
shapiro.test(finmultilinearregred$residuals) #p.value < 0.05 SO WE
DONT HAVE NORMALITY IN THE RESIDUALS
jb.norm.test(finmultilinearregred$residuals) #p.value < 0.05 SO WE
DONT HAVE NORMALITY IN THE RESIDUALS
ad.test(finmultilinearregred$residuals)  #p.value < 0.05 SO WE DONT
HAVE NORMALITY IN THE RESIDUALS
cvm.test(finmultilinearregred$residuals) #p.value < 0.05 SO WE DONT
HAVE NORMALITY IN THE RESIDUALS
ks.test(finmultilinearregred$residuals,"pnorm",
mean(finmultilinearregred$residuals),
sd(finmultilinearregred$residuals) ) #p.value < 0.05 SO WE DONT HAVE
NORMALITY IN THE RESIDUALS
res.student <- rstudent(finmultilinearregred)
res.student
plot(res.student,pch= 15, cex= .5,ylab = "Studentized Residuals")
abline(h=c(-2,0,2), lty= c(2,1,2))
res.standard <- rstandard(finmultilinearregred)
res.standard
plot(res.standard, phc = 15, cex = .5,ylab = "Standarized Residuals")
abline(h=c(-2,0,2), lty= c(2,1,2))
# No3 WE CHECK FOR HETEROSCEDASTICITY IN THE RESIDUALS OF THE FINAL
REGRESSION
bptest(finmultilinearregred)#P.VALUE < 0.05 SO WE HAVE
HETEROSKEDASTICITY
# No4 WE CHECK FOR LINEARITY
plot(finmultilinearregred$residuals,main = "Multiple Regression
Residuals")
# No5 WE CHECK FOR CORRELATION BETWEEN X VARIABLES AND THE RESIDUALS
cor.test(mydatasetred1$alcohol, finmultilinearregred$residuals)
cor.test(mydatasetred1$volatile.acidity,
finmultilinearregred$residuals)
cor.test(mydatasetred1$chlorides, finmultilinearregred$residuals)
cor.test(mydatasetred1$total.sulfur.dioxide,
finmultilinearregred$residuals)
cor.test(mydatasetred1$pH, finmultilinearregred$residuals)
cor.test(mydatasetred1$sulphates, finmultilinearregred$residuals)
# (6) WE CHECK FOR AUTOCORRELATION
durbinWatsonTest(multilinearregred) #P.VALUE < 0.05 SO WE HAVE
AUTOCORRELATION
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
```

```
# ROW INDICES FOR TRAINING DATA (70%-30%)
trainingRowIndexred<- sample(1:nrow(mydatasetred1),
0.7*nrow(mydatasetred1))
training.setred <- mydatasetred1[trainingRowIndexred, ]
testing.setred  <- mydatasetred1[-trainingRowIndexred, ]
# PREDICTION METRICS
multi_regred <- lm(training.setred$quality ~
training.setred$volatile.acidity+training.setred$chlorides+training.s
etred$total.sulfur.dioxide+training.setred$pH+training.setred$sulphat
es+training.setred$alcohol, data = training.setred)
summary(multi_regred)
predicted_valuered <- predict(multi_regred , newdata =
testing.setred)
predicted_valuered
# ML metrics
# RMSE
RMSE(predicted_valuered, testing.setred$quality)
# MAE
MAE(predicted_valuered, testing.setred$quality)
# MAPE
MAPE(predicted_valuered, testing.setred$quality)
# MSE
MSE(testing.setred$quality, predicted_valuered)
# MIN - MAX ACCURACY
dist_predred <- predict(multi_regred , testing.setred)
actuals_predsred <- data.frame(cbind(actuals=testing.setred$quality ,
predicteds=dist_predred))
min_max_accuracyred <-
mean(apply(actuals_predsred,1,min)/apply(actuals_predsred,1,max))
min_max_accuracyred
# CROSS-VALIDATION (10 FOLD)
set.seed(123)
model<- train(quality~., data = mydatasetred1, method = "lm",
trControl = trainControl(method = "cv", number = 10, verboseIter =
TRUE))
model
```

Ridge Regression Red Dataset

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("glmnet")
install.packages("ISLR")
library(ISLR)
library(glmnet)
install.packages("DMwR")
library(DMwR)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE CREATE RIDGE REGRESSION (WARNINR: IN RIDGE REGRESSION WE DO NOT
CHECK FOR STATISTICALLY SIGNIFICANT VARIABLES IN ORDER TO CREATE THE
REGRESSION LINE. RIDGE CONTAINS EVERY SINGLE INDEPENDENT VARIABLE
THAT EXISTS IN THE DATASET)
x<-model.matrix(quality~.,mydatasetred1)[,-1]
y<-na.omit(mydatasetred1$quality)
```

```
grid<-10^ seq(10,-2, length=100) # we create a sequence for the
lambda that we are going to use (ranging from a lambda = 10^10 to a
lambda = 10^-2)
ridge.mod<-glmnet(x,y,alpha = 0,lambda = grid) # alpha=0 (ridge)
ridge.mod
predict(ridge.mod,s=50,type = "coefficients")[1:12,] # we take the
coef if we use a lambda=50 for example(this is not mandatory)
# WE CREATE TRAINING AND TESTING SET
set.seed(1)
train<-sample(1:nrow(x),nrow(x)/2)
test<-(-train)
y.test<-y[test]
# WE FIND OUT WHICH IS THE BEST LAMBDA FOR OUR REGRESSION WITH CROSS-
VALIDATION
set.seed(1)
cv.out<-cv.glmnet(x[train,],y[train],alpha=0)
plot(cv.out)
bestlambda<-cv.out$lambda.min
bestlambda #the bestlambdafor our model is lambda=0.7940699
# WE PREDICT AGAIN ON THE TESTING SET (WITH THE BEST LAMBDA) AND WE
GET THE PREDICTION METRICS
ridge.pred<-predict(ridge.mod,s=bestlambda,newx = x[test,])
mean((ridge.pred-y.test)^2) #this is our MSE
regr.eval(trues = y.test, preds = ridge.pred)
# FINALLY WE RUN AGAIN THE REGRESSION WITH THE WHOLE DATASET AND THE
BEST LAMBDA AND WE HAVE ITS COEFFICIENTS
out=glmnet(x,y,alpha = 0)
ridge.coef<-predict(out,type = "coefficients",s=bestlambda)[1:12,]
ridge.coef
```

Lasso Regression Red Dataset

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("glmnet")
install.packages("ISLR")
library(ISLR)
library(glmnet)
install.packages("DMwR")
library(DMwR)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE CREATE LASSO REGRESSION (WARNINR: IN LASSO REGRESSION WE DO NOT
CHECK FOR STATISTICALLY SIGNIFICANT VARIABLES IN ORDER TO CREATE THE
REGRESSION LINE. LASSO CONTAINS EVERY SINGLE INDEPENDENT VARIABLE
THAT EXISTS IN THE DATASET)
x<-model.matrix(quality~.,mydatasetred1)[,-1]
y<-na.omit(mydatasetred1$quality)
grid<-10^ seq(10,-2, length=100) # we create a sequence for the
lambda that we are going to use (ranging from a lambda = 10^10 to a
lambda = 10^-2)
lasso.mod<-glmnet(x,y,alpha = 1,lambda = grid) # alpha=1 (lasso)
lasso.mod
plot(lasso.mod, xlab = "λ")
legend("bottomleft", lwd = 1, col = 1:11, legend = colnames(x), cex =
.3)
```

```
# WE CREATE TRAINING AND TESTING SET
set.seed(1)
train<-sample(1:nrow(x),nrow(x)/2)
test<-(-train)
y.test<-y[test]
# WE FIND OUT WHICH IS THE BEST LAMBDA FOR OUR REGRESSION WITH CROSS-
VALIDATION
set.seed(1)
cv.out<-cv.glmnet(x[train,],y[train],alpha=1)
plot(cv.out)
bestlambda<-cv.out$lambda.min
bestlambda #the best lambda for our model is lambda=0.01921751
# WE PREDICT ON THE TESTING SET (WITH THE BEST LAMBDA)
lasso.pred<-predict(lasso.mod,s=bestlambda,newx=x[test,])
mean((lasso.pred-y.test)^2) #this is our MSE
regr.eval(trues = y.test, preds = lasso.pred) #here we have all the
metrics
# FINALLY WE RUN AGAIN THE REGRESSION WITH THE WHOLE DATASET AND THE
BEST LAMBDA AND WE HAVE ITS COEFFICIENTS
out=glmnet(x,y,alpha = 1,lambda=grid)
lasso.coef<-predict(out,type = "coefficients",s=bestlambda)[1:12,]
lasso.coef
lasso.coef[lasso.coef!=0]
```

Polynomial Regression Red Dataset

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("MLmetrics")
library(MLmetrics)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE BUILD THE SIMPLE REGRESSION AND THEN WE BUILD FOUR DIFFERENT
POLYNOMIAL REGRESSIONS
fit<- lm(mydatasetred1$quality ~ mydatasetred1$alcohol)
plot(mydatasetred1$quality ~ mydatasetred1$alcohol,
main="Quality~Alcohol", xlab="Alcohol", ylab="Quality")
abline(fit, col="red")
fit2 <- lm(mydatasetred1$quality ~ mydatasetred1$alcohol +
I(mydatasetred1$alcohol^2))
fit3 <- lm(mydatasetred1$quality ~ mydatasetred1$alcohol +
I(mydatasetred1$alcohol^2) +I(mydatasetred1$alcohol^3))
fit4 <- lm(mydatasetred1$quality ~ mydatasetred1$alcohol +
I(mydatasetred1$alcohol^2) +I(mydatasetred1$alcohol^3) +
I(mydatasetred1$alcohol^4))
fit5 <- lm(mydatasetred1$quality ~ mydatasetred1$alcohol +
I(mydatasetred1$alcohol^2) +I(mydatasetred1$alcohol^3) +
I(mydatasetred1$alcohol^4) + I(mydatasetred1$alcohol^5))
summary(fit)
summary(fit2)
summary(fit3)
summary(fit4)
summary(fit5)
# WE FIND OUT WHICH IS THE MOST APPROPRIATE DEGREE FOR OUR REGRESSION
anova(fit, fit2, fit3, fit4, fit5) #fit3 is the best form of
polynomial for our analysis
```

```
# WE ADD THE POLYNOMIAL REGRESSION (FIT3) INTO THE PREVIOUS PLOT
lines(smooth.spline(mydatasetred1$alcohol, predict(fit3)),
col="blue", lwd=3)
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
# ROW INDICES FOR TRAINING DATA (70%-30%)
trainingRowIndexred<- sample(1:nrow(mydatasetred1),
0.7*nrow(mydatasetred1))
training.setred <- mydatasetred1[trainingRowIndexred, ]
testing.setred  <- mydatasetred1[-trainingRowIndexred, ]
# PREDICTION METRICS
fit3.prediction <- lm(training.setred$quality ~
training.setred$alcohol + I(training.setred$alcohol^2)
+I(training.setred$alcohol^3))
predicted_valuered <- predict(fit3.prediction , newdata =
testing.setred)
predicted_valuered
# RMSE
RMSE(predicted_valuered, testing.setred$quality)
# MAE
MAE(predicted_valuered, testing.setred$quality)
# MAPE
MAPE(predicted_valuered, testing.setred$quality)
# MSE
MSE(testing.setred$quality, predicted_valuered)
# MIN - MAX ACCURACY
dist_predred <- predict(fit3.prediction , testing.setred)
actuals_predsred <- data.frame(cbind(actuals=testing.setred$quality ,
predicteds=dist_predred))
min_max_accuracyred <-
mean(apply(actuals_predsred,1,min)/apply(actuals_predsred,1,max))
min_max_accuracyred
```

Logistic Regression Red Dataset

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("tidyverse")
library(tidyverse)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE REMOVE QUALITY
newdataset<-mydatasetred1[ ,1:11]
# WE CREATE A FACTOR VARIABLE FOR QUALITY
factor.quality<-cut(mydatasetred1$quality, breaks = c(0,5,10), labels
= c("Bad Quality", "Good Quality"))
# WE ADD THE VARIABLE IN THE EXISTING DATASET
newdataset<-cbind(newdataset, factor.quality)
summary(newdataset)
# WE CREATE THE LOGISTIC MODEL
glm.fits<-
glm(factor.quality~newdataset$fixed.acidity+newdataset$volatile.acidi
ty+newdataset$citric.acid+newdataset$residual.sugar+newdataset$chlori
des+newdataset$free.sulfur.dioxide+newdataset$total.sulfur.dioxide+ne
```

```
wdataset$density+newdataset$pH+newdataset$sulphates+newdataset$alcoho
l, data = newdataset, family = binomial)
summary(glm.fits)
coef(glm.fits)
# PREDICTION ON THE WHOLE DATASET AND EVALUATION METRIC
glm.probs<-predict(glm.fits, newdataset,type="response")
glm.probs[1:10] #these are the first 10 probabilities
glm.probs.rd <- ifelse(glm.probs > 0.5, 1, 0)
table(glm.probs.rd, newdataset[,12])
accuracy<- table(glm.probs.rd, newdataset[,12])
sum(diag(accuracy))/sum(accuracy)
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
# ROW INDICES FOR TRAINING DATA (70%-30%)
data2 = sort(sample(nrow(newdataset), nrow(newdataset)*.7))
train<- newdataset[data2,]
test<- newdataset[-data2,]
dim(train)
dim(test)
# PREDICTION ON TESTING SET AND EVALUATION METRIC
glm.fits2 <- glm(factor.quality ~., data = train, family =
binomial(link = "logit"))
glm.probs2 <- predict(glm.fits2, test, type="response")
glm.probs2[1:10] #these are the first 10 probabilities
glm.probs2.rd <- ifelse(glm.probs2 > 0.5, 1, 0)
table(glm.probs2.rd, test[,12])
accuracy<- table(glm.probs2.rd, test[,12])
sum(diag(accuracy))/sum(accuracy)
# WE BUILD AGAIN THE LOGISTIC REGRESSION, THIS TIME WITHOUT THE
INSIGNIFICANT VARIABLES
newdataset2 <- select(newdataset, -
one_of('fixed.acidity','citric.acid', 'residual.sugar',
                                'pH','density'))
glm.fits3<-
glm(factor.quality~newdataset2$volatile.acidity+newdataset2$chlorides
+newdataset2$free.sulfur.dioxide+newdataset2$total.sulfur.dioxide+new
dataset2$sulphates+newdataset2$alcohol, data = newdataset2, family =
binomial)
summary(glm.fits3)
coef(glm.fits3)
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
# ROW INDICES FOR TRAINING DATA (70%-30%)
data3 = sort(sample(nrow(newdataset2), nrow(newdataset2)*.7))
train2 <- newdataset2[data3,]
test2 <- newdataset2[-data3,]
dim(train2)
dim(test2)
# PREDICTION ON TESTING SET AND EVALUATION METRIC
glm.fits4<-glm(factor.quality~., data = train2, family = binomial)
glm.probs4<-predict(glm.fits4, test2,type="response")
glm.probs4[1:10] #these are the first 10 probabilities
glm.probs4.rd <- ifelse(glm.probs4 > 0.5, 1, 0)
table(glm.probs4.rd, test2[ ,7])
accuracy<- table(glm.probs4.rd, test2[,7])
sum(diag(accuracy))/sum(accuracy)
```

Poisson Regression Red Dataset

```
#WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("qcc")
library(qcc)
install.packages("MLmetrics")
library(MLmetrics)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE CREATE POISSON REGRESSION
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$fixed.acidity+mydatasetred1$v
olatile.acidity+mydatasetred1$citric.acid+mydatasetred1$residual.suga
r+mydatasetred1$chlorides+mydatasetred1$free.sulfur.dioxide+mydataset
red1$total.sulfur.dioxide+mydatasetred1$density+mydatasetred1$pH+myda
tasetred1$sulphates+mydatasetred1$alcohol, family = poisson(link =
"log"))
summary(poisson.model)
coef<-round(coef(poisson.model),3)
coef #but in this form our coefficients are in log formation
round(exp(coef(poisson.model)),3) #with this function we interpret
the regression coefficients in the original scale of the dependent
variable (not in the log one)
# WE THROW AWAY THE INSIGNIFICANT INDEPENDENT VARIABLES
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$fixed.acidity+mydatasetred1$v
olatile.acidity+mydatasetred1$citric.acid+mydatasetred1$residual.suga
r+mydatasetred1$chlorides+mydatasetred1$free.sulfur.dioxide+mydataset
red1$total.sulfur.dioxide+mydatasetred1$pH+mydatasetred1$sulphates+my
datasetred1$alcohol, family = poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$citric.acid+mydatasetred1$residual.sugar+mydatasetred1$chlorides+my
datasetred1$free.sulfur.dioxide+mydatasetred1$total.sulfur.dioxide+my
datasetred1$pH+mydatasetred1$sulphates+mydatasetred1$alcohol, family
= poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$citric.acid+mydatasetred1$chlorides+mydatasetred1$free.sulfur.dioxi
de+mydatasetred1$total.sulfur.dioxide+mydatasetred1$pH+mydatasetred1$
sulphates+mydatasetred1$alcohol, family = poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$chlorides+mydatasetred1$free.sulfur.dioxide+mydatasetred1$total.sul
fur.dioxide+mydatasetred1$pH+mydatasetred1$sulphates+mydatasetred1$al
cohol, family = poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$chlorides+mydatasetred1$total.sulfur.dioxide+mydatasetred1$pH+mydat
asetred1$sulphates+mydatasetred1$alcohol, family = poisson(link =
"log"))
```

```
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$chlorides+mydatasetred1$free.sulfur.dioxide+mydatasetred1$total.sul
fur.dioxide+mydatasetred1$sulphates+mydatasetred1$alcohol, family =
poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$chlorides+mydatasetred1$total.sulfur.dioxide+mydatasetred1$pH+mydat
asetred1$sulphates+mydatasetred1$alcohol, family = poisson(link =
"log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$chlorides+mydatasetred1$total.sulfur.dioxide+mydatasetred1$sulphate
s+mydatasetred1$alcohol, family = poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$total.sulfur.dioxide+mydatasetred1$sulphates+mydatasetred1$alcohol,
family = poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$sulphates+mydatasetred1$alcohol, family = poisson(link = "log"))
summary(poisson.model)
poisson.model<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$alcohol, family = poisson(link = "log"))
summary(poisson.model)
# WE KEEP TWO VARIABLES (VOLATILE ACIDITY AND ALCOHOL)
# WE CREATE POISSON REGRESSION (THIS TIME WITHOUT THE INSIGNIFICANT
VARIABLES)
poisson.model2<-
glm(mydatasetred1$quality~mydatasetred1$volatile.acidity+mydatasetred
1$alcohol, family = poisson(link = "log"))
summary(poisson.model2)
coef<-round(coef(poisson.model2),3)
coef #but in this form our coefficients are in log formation
round(exp(coef(poisson.model2)),3) #with this function we interpret
the regression coefficients in the original scale of the dependent
variable (not in the log one)
# WE SHOULD CHECK FOR OVERDISPERSION (IT OCCURS IN POISSON REGRESSION
BECAUSE THE VARIANCE AND MEANS ARE EQUAL)
qcc.overdispersion.test(mydatasetred1$quality, type = "poisson")
#P.VALUE>0.05 SO THERE IS NO OVERDISPERSION WHICH IS GOOD
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
# ROW INDICES FOR TRAINING DATA (70%-30%)
trainingRowIndexred<- sample(1:nrow(mydatasetred1),
0.7*nrow(mydatasetred1))
training.setred <- mydatasetred1[trainingRowIndexred, ]
testing.setred  <- mydatasetred1[-trainingRowIndexred, ]
# PREDICTION ON TESTING SET AND EVALUATION METRIC
poisson.model3<-
glm(training.setred$quality~training.setred$volatile.acidity+training
.setred$alcohol, family = poisson(link = "log"))
summary(poisson.model3)
```

```
predicted_valuered <- predict(poisson.model3 , newdata =
testing.setred)
predicted_valuered
# ML metrics
#RMSE
RMSE(predicted_valuered, testing.setred$quality)
# MAE
MAE(predicted_valuered, testing.setred$quality)
# MAPE
MAPE(predicted_valuered, testing.setred$quality)
# MSE
MSE(testing.setred$quality, predicted_valuered)
# MIN - MAX ACCURACY
dist_predred <- predict(poisson.model3 , testing.setred)
actuals_predsred <- data.frame(cbind(actuals=testing.setred$quality ,
predicteds=dist_predred))
min_max_accuracyred <-
mean(apply(actuals_predsred,1,min)/apply(actuals_predsred,1,max))
min_max_accuracyred
```

Decision Tree Regression Red Dataset

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("rpart.plot")
library(rpart.plot)
install.packages("rpart")
library(rpart)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE REMOVE QUALITY
newdataset<-mydatasetred1[ ,1:11]
# WE CREATE A FACTOR VARIABLE FOR QUALITY
factor.quality<-cut(mydatasetred1$quality, breaks = c(0,5,10), labels
= c("Bad Quality", "Good Quality"))
# WE ADD THE VARIABLE IN THE EXISTING DATASET
newdataset<-cbind(newdataset, factor.quality)
summary(newdataset)
# WE SHUFFLE THE DATASET
shuffle_index <- sample(1:nrow(newdataset))
head(shuffle_index)
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
# ROW INDICES FOR TRAINING DATA (70%-30%)
data2 = sort(sample(nrow(newdataset), nrow(newdataset)*.7))
train<- newdataset[data2,]
test<- newdataset[-data2,]
dim(train)
dim(test)
# WE BUILD THE DECISION TREE MODEL
fit<- rpart(factor.quality~., data = train, method = 'class')
summary(fit)
fitrpart.plot(fit, extra = 106)
# PREDICTION ON TESTING SET AND EVALUATION METRIC
predict_unseen <-predict(fit, test, type = 'class')
```

```
table_mat <- table(test$factor.quality, predict_unseen)
table_mat
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
print(paste('Accuracy for test', accuracy_Test))
```

Random Forest Regression Red Dataset

```
# WE DOWNLOAD THE APPROPRIATE PACKAGES
install.packages("randomForest")
library(randomForest)
install.packages("caret")
library(caret)
install.packages("e1071")
library(e1071)
# WE IMPORT THE DATASETS
mydatasetred1 <- read.csv("winequality-red.csv", sep = ";", dec =
".", header = T)
summary(mydatasetred1)
View(mydatasetred1)
str(mydatasetred1)
# WE REMOVE QUALITY
newdataset<-mydatasetred1[ ,1:11]
# WE CREATE A FACTOR VARIABLE FOR QUALITY
factor.quality<-cut(mydatasetred1$quality, breaks = c(0,5,10), labels
= c("Bad Quality", "Good Quality"))
# WE ADD THE VARIABLE IN THE EXISTING DATASET
newdataset<-cbind(newdataset, factor.quality)
summary(newdataset)
# WE CREATE TRAINING AND TESTING SET
# SETTING SEED TO REPRODUCE RESULTS OF RANDOM SAMPLING
set.seed(123)
# ROW INDICES FOR TRAINING DATA (70%-30%)
data2 = sort(sample(nrow(newdataset), nrow(newdataset)*.7))
train<- newdataset[data2,]
test<- newdataset[-data2,]
dim(train)
dim(test)
# TRAIN THE MODEL
# DEFINE THE CONTROL
trControl<- trainControl(method = "cv",number = 10,search = "grid")
set.seed(1234)
# RUN THE RANDOM FOREST FUNCTION IN ORDER TO FIND OUT THE BEST mtry
THAT GIVES US THE BEST PREDICTION ACCURACY
set.seed(1234)
tuneGrid<- expand.grid(.mtry = c(1: 10))
rf_mtry <- train(factor.quality~.,
data = train,
method = "rf",
metric = "Accuracy",
tuneGrid = tuneGrid,
trControl = trControl,
importance = TRUE,
nodesize = 14,
ntree = 300)
print(rf_mtry)
rf_mtry$bestTune$mtry
max(rf_mtry$results$Accuracy)
best_mtry <- rf_mtry$bestTune$mtry
best_mtry
```

```
# SEARCH AND FIND OUT THE BEST MAXNODES
store_maxnode <- list()
tuneGrid<- expand.grid(.mtry = best_mtry)
for (maxnodes in c(5: 15)) {
set.seed(1234)
  rf_maxnode <- train(factor.quality~.,
data = train,
method = "rf",
metric = "Accuracy",
tuneGrid = tuneGrid,
trControl = trControl,
importance = TRUE,
nodesize = 14,
maxnodes = maxnodes,
ntree = 300)
  current_iteration <- toString(maxnodes)
  store_maxnode[[current_iteration]] <- rf_maxnode
}
results_mtry <- resamples(store_maxnode)
summary(results_mtry)
# SEARCH AND FIND OUT THE BEST NTREE
store_maxtree <- list()
for (ntree in c(250, 300, 350, 400, 450, 500, 550, 600, 800, 1000,
2000)) {
set.seed(5678)
  rf_maxtree <- train(factor.quality~.,
data = train,
method = "rf",
metric = "Accuracy",
tuneGrid = tuneGrid,
trControl = trControl,
importance = TRUE,
nodesize = 14,
maxnodes = 5,
ntree = ntree)
key<- toString(ntree)
  store_maxtree[[key]] <- rf_maxtree
}
results_tree <- resamples(store_maxtree)
summary(results_tree)
# WE BUILD THE MODEL WITH THE FINAL MAXNODES, NTREE AND MTRY
fit_rf <- train(factor.quality~.,
train,
method = "rf",
metric = "Accuracy",
tuneGrid = tuneGrid,
trControl = trControl,
importance = TRUE,
nodesize = 14,
ntree = 350,
maxnodes = 5)
# TEST AND EVALUATE THE MODEL
prediction<-predict(fit_rf,test)
confusionMatrix(prediction, test$factor.quality)
```

References

Regunathan Radhakrishnan, Daniel Nikovski, Kadir Peker and Ajay Divakaran. (2006*). A Comparison between Polynomial and Locally Weighted Regression for Fault Detection and Diagnosis of HVAC Equipment* [Conference presentation]. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics. https://ieeexplore.ieee.org/document/4153046/authors#authors

Acharya, M. S., Armaan, A., & Antony, A. S. (2019). *A Comparison of Regression Models for Prediction of Graduate Admissions* [Conference presentation]. 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 1–5. https://ieeexplore.ieee.org/document/8862140

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R* (7th ed.). USA: Springer International Publishing.

Härdle, W. K., & Simar, L. (2019). *Applied Multivariate Statistical Analysis* (5th ed.). USA: Springer International Publishing.

Analytics Vidhya Team. (2016, July 16). Going Deeper into Regression Analysis with Assumptions, Plots & Solutions. *Analytics Vidhya*. https://libanswers.snhu.edu/faq/190823

Booth, A., Gerding, E., & McGroarty, F. (2014). Automated trading with performance weighted random forest and seasonality. *Expert Systems with Applications*, *41*(8), 3651–3661. https://www.sciencedirect.com/science/article/pii/S0957417413009731

Feng, J., Wang, Y., Peng, J., Sun, M., Zeng, J., & Jiang, H. (2019). Comparison between logistic regression and machine learning algorithms on survival prediction of traumatic brain injuries. *Journal of Critical Care*, *54*, 110–116. https://www.sciencedirect.com/science/article/pii/S0883944119302618

Nie, G., Rowe, W., Zhang, L., Tian, Y., & Shi, Y. (2011). Credit card churn forecasting by logistic regression and decision tree. *Expert Systems with Applications*, *38*(12), 15273–15285. https://www.sciencedirect.com/science/article/pii/S0957417411009237

Kafazi, I. E., Bannari, R., Abouabdellah, A., Aboutafail, M. O., & Guerrero, J. M. (2017). Energy Production: A Comparison of Forecasting Methods using the Polynomial Curve Fitting and Linear Regression. *2017 International Renewable and Sustainable Energy Conference (IRSEC)*, 1–5. https://ieeexplore.ieee.org/document/8477278

Kaytez, F., Taplamacioglu, M. C., Cam, E., & Hardalac, F. (2015). Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power & Energy Systems*, *67*, 431–438. https://www.sciencedirect.com/science/article/pii/S0142061514007637

Dataquest Team. (2019, February 27). Tutorial: Poisson Regression in R. *Dataquest.* https://www.dataquest.io/blog/tutorial-poisson-regression-in-r/

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, *35*(5), 352–359. https://www.sciencedirect.com/science/article/pii/S1532046403000340

Pagliara, F., & Mauriello, F. (2020). Modelling the impact of High Speed Rail on tourists with Geographically Weighted Poisson Regression. *Transportation Research Part A: Policy and Practice*, *132*, 780–790. https://www.sciencedirect.com/science/article/pii/S0965856418308851

Exterkate, P., Groenen, P. J. F., Heij, C., & van Dijk, D. (2016). Nonlinear forecasting with many predictors using kernel ridge regression. *International Journal of Forecasting*, *32*(3), 736–753. https://www.sciencedirect.com/science/article/pii/S0169207016000182

Larivière, B., & Van den Poel, D. (2005). Predicting customer retention and profitability by using random forest and regression forest techniques. *Expert Systems with Applications*, *29*(2), 472–484. https://www.sciencedirect.com/science/article/pii/S0957417405000965

Tso, G. K. F., & Yau, K. K. W. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, *32*(9), 1761–1768. https://www.sciencedirect.com/science/article/pii/S0360544206003288

Mayooran Thevaraja, Azizur Rahman and Mathew Gabirial. (2019). *Recent Developments in Data Science: Comparing Linear, Ridge and Lasso Regressions Techniques Using Wine Data* [Conference presentation]. International Conference on Digital Image and Signal Processing 2019. https://www.researchgate.net/publication/324870033_Recent_Developments_in_Data_Science_Comparing_Linear_Ridge_and_Lasso_Regressions_Techniques_Using_Wine_Data

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288. JSTOR. https://www.jstor.org/stable/2346178?seq=1#metadata_info_tab_contents

Edward C.Malthouse. (1999). Ridge regression and direct marketing scoring models. *Journal of Interactive Marketing*, 13, no. 4 (January 1, 1999): 10–23. https://www.sciencedirect.com/science/article/pii/S1094996899702446

Gaurav Bansal. (2011). What are the four assumptions of linear regression. *Gaurav Bansal Blog*. https://blog.uwgb.edu/bansalg/statistics-data-analytics/linear-regression/what-are-the-four-assumptions-of-linear-regression/

Yao, Y., Xu, B., & He, J. (2017). Wine Evaluation Modeling Based on Lasso and Support Vector Regression. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *21*(6), 998–1008. https://www.fujipress.jp/jaciii/jc/jacii002100060998/

Athanasiadis.I & Ioannides.D . (2017). *Στατιστική και μηχανική μάθηση με την R: Θεωρία και εφαρμογές.* Ελλάδα: Εκδόσεις Τζιόλα.